

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra Elektroniky

Hardwarový terminál pro zpracování dat automobilové komunikační
sběrnice

Hardware Terminal for Data processing of Vehicle Communication
Bus

2010

Tomáš PSZCZÓŁKA

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Zadání diplomové práce

Student: **Bc. Tomáš Pszczółka**
Studijní program: **N2649 Elektrotechnika**
Studijní obor: **2612T015 Elektronika**
Téma: **Hardwarový terminál pro zpracování dat automobilové komunikační sběrnice**
Hardware Terminal for Data processing of Vehicle Communication Bus

Zásady pro vypracování:

1. Proveďte detailní rešerši z oblasti autodiagnostických komunikačních vedeních - typy, principy a vlastnosti.
2. Dle pokynů vedoucího diplomové práce realizujte hardwarový terminál pro zpracování dat sériové komunikační sběrnice vytypovaného osobního vozidla.
3. Sestavte typyzovanou laboratorní úlohu s využitím realizovaného zařízení a vypracujte vzorový protokol.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Petr Šimoník, Ph.D.**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



Recy

doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry

Tomáš

prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 07. 05. 2010

.....

Plné jméno

Poděkování

Děkuji vedoucímu diplomové práce Ing. Petru Šimoníkovi, Ph.D. za odborné konzultace při tvorbě této práce. Také děkuji svým rodičům, bratrovi a sestře za podporu při studiu.

Abstrakt

Cílem této diplomové práce je detailní popis typů a principů automobilových komunikačních vedení. Dalším úkolem práce je návrh a realizace hardwarového terminálu pro zpracování dat sériové komunikační sběrnice. Dále tato práce uvádí typovou laboratorní úlohu s realizovaným hardwarem. V přílohách je uvedeno schéma, včetně návrhu desky plošných spojů, dále je zde umístěn vypracovaný vzorový protokol. Další přílohy jsou doplňujícími teoretickými poznatky k vlastní diplomové práci.

Klíčová slova

CAN sběrnice, LIN sběrnice, FlexRay sběrnice, K – vedení, KWP 2000, KWP 1281, TP 1.6, TP 2.0.

Abstract

The aim of this diploma thesis is detail description of types of automotive communication lines. Next task of thesis is design and realization of hardware terminal for serial communication bus data processing. Next, this thesis provides benchmark with realized hardware. In appendix there are wiring scheme including design of printed circuit board, next, there is located elaborated model lab report. Next appendixes are additional theoretical knowledge to thesis itself.

Keywords

CAN Bus, LIN Bus, FlexRay Bus, K – Line, KWP 2000, KWP 1281, TP 1.6, TP 2.0.

Obsah

1	ÚVOD	1
2	SBĚRNICE CAN (CONTROLLER AREA NETWORK)	3
2.1	HISTORIE VZNIKU CAN	3
2.2	ZÁKLADNÍ VLASTNOSTI CAN SBĚRNICE	4
2.2.1	<i>Princip komunikace.....</i>	<i>5</i>
2.3	PARAMETRY CAN SBĚRNICE - OBECNĚ	5
2.4	VRSTVY CAN SBĚRNICE	6
2.5	FYZICKÝ VRSTVA (PŘENOSOVÉ MÉDIUM).....	7
2.5.1	<i>Normy fyzické vrstvy.....</i>	<i>8</i>
2.5.2	<i>Zamezení rušů na sběrnici CAN.....</i>	<i>12</i>
2.6	LINKOVÁ VRSTVA.....	13
2.6.1	<i>Řízení přístupu k médium.....</i>	<i>13</i>
2.6.2	<i>Časování a synchronizace bitů.....</i>	<i>14</i>
2.6.3	<i>Zabezpečení přenášených dat a detekce chyb.....</i>	<i>17</i>
2.6.4	<i>Chybové stavy uzlů (signalizace chyb).....</i>	<i>18</i>
2.6.5	<i>Filtrování zpráv.....</i>	<i>19</i>
2.7	FORMÁT RÁMCE CAN PROTOKOLU	20
2.7.1	<i>Formát datového rámce.....</i>	<i>21</i>
2.7.2	<i>Formát rámce žádosti o data podle CAN 2.0A (B).....</i>	<i>23</i>
2.7.3	<i>Formát chybového rámce.....</i>	<i>24</i>
2.7.4	<i>Formát rámce o přetížení</i>	<i>24</i>
2.8	HARDWAROVÉ ŘEŠENÍ JEDNOTKY CAN SBĚRNICE	25
3	LIN SBĚRNICE (LOCAL INTERCONNECT NETWORK)	28
3.1	HISTORIE LIN SBĚRNICE.....	28
3.2	ZÁKLADNÍ VLASTNOSTI LIN SBĚRNICE.....	29
3.3	PRINCIP KOMUNIKACE	30
3.3.1	<i>Směry komunikace</i>	<i>30</i>
3.4	FYZICKÁ VRSTVA	32
3.4.1	<i>Zamezení rušů na sběrnici LIN.....</i>	<i>34</i>
3.5	LINKOVÁ VRSTVA.....	34
3.5.1	<i>Formát rámce sběrnice LIN.....</i>	<i>35</i>

3.5.2	<i>Mezera zprávy – synchronizační pauza</i>	36
3.5.3	<i>Synchronizační pole</i>	37
3.5.4	<i>Pole identifikátoru</i>	37
3.5.5	<i>Datový rámec</i>	38
3.5.6	<i>Pole kontrolního součtu</i>	39
3.6	NĚKTERÉ DALŠÍ FUNKCE	40
3.6.1	<i>Oživení komunikace (Wake Up)</i>	40
3.6.2	<i>Uspání (Sleep mód)</i>	41
3.7	HARDWAROVÉ ŘEŠENÍ JEDNOTKY LIN SBĚRNICE	41
4	SBĚRNICE FLEXRAY	43
4.1	HISTORIE SBĚRNICE FLEXRAY	43
4.2	VLASTNOSTI SBĚRNICE FLEXRAY	43
4.3	FYZICKÁ VRSTVA	44
4.3.1	<i>Zamezení rušení na sběrnici FlexRay (EMC)</i>	45
4.3.2	<i>Topologie sítě</i>	46
4.4	LINKOVÁ VRSTVA	47
4.4.1	<i>Komunikační cyklus standardu</i>	47
4.4.2	<i>Mechanismus Synchronizace jednotek</i>	49
4.4.3	<i>Mechanismus startu sítě</i>	51
4.4.4	<i>Formát komunikačního rámce</i>	51
4.5	HARDWAROVÉ ŘEŠENÍ JEDNOTKY - FLEXRAY SBĚRNICE	53
5	VEDENÍ K – LINE, L - LINE (ISO 9141)	55
5.1	INICIALIZACE KOMUNIKACE	56
5.2	DATOVÝ RÁMEC	58
6	PŘEHLED OSTATNÍCH VEDENÍ	59
6.1	DATOVÁ SBĚRNICE MOST	59
6.2	DATOVÁ SBĚRNICE ACP	59
6.3	SAE J1850	59
7	DIAGNOSTICKÉ PROTOKOLY	60
7.1	TRANSPORTNÍ VRSTVA	60
7.1.1	<i>VWTP 1.6</i>	61
7.1.2	<i>VWTP 2.0</i>	62

7.2	APLIKAČNÍ VRSTVA.....	64
7.2.1	<i>KW 1281</i>	64
7.2.2	<i>KWP 2000 (ISO 14230)</i>	65
8	REALIZACE HARDWAROVÉHO TERMINÁLU (K – LINE INTERFACE).....	66
8.1	PŘEVODNÍK RS 232 / TTL (TTL / RS 232).....	66
8.2	PŘEVODNÍK K, L LINE / TTL (TTL / K, L LINE)	67
8.2.1	<i>Popis obvodu SI9243AEY</i>	67
8.2.2	<i>Zásuvka EOB</i>	68
8.3	OVĚŘENÍ FUNKČNOSTI HARDWARE	69
9	ZÁVĚR	70
	SEZNÁM POŽITÉ LITERATURY.....	71
	SEZNAM PŘÍLOH	74

1 Úvod

Automobilové komunikační sběrnice jsou v dnešní době stále aktuálnější. Vývojem inteligentních automobilových systému a mechanismu je potřeba neustále zdokonalovat automobilové sběrnice struktury, jak v oblasti přenosové rychlosti, spolehlivosti, úrovně zabezpečení či také ceny.

Efektivní využití komunikačních vedení automobilu vyžaduje detailní znalost jednotlivých komunikačních systému. Snahou práce bylo zachytit nejdůležitější podstatné z dnešních používaných vedení s ohledem na rozsah práce a odbornost témat. Proto jsou kapitoly doplněny odkazy na literatury, kde je možno získat ještě detailnější informace, např. z hlediska návrhu jednotek atd.

V první části rešerše je popsána CAN sběrnice, která pracuje s dostatečnou spolehlivostí a přenosovou rychlostí. Je využívána v oblasti automobilové diagnostiky a komunikace jednotek v automobilu.

Další kapitola popisuje LIN sběrnici, která vznikla z důvodu rostoucího nasazení řídicích jednotek připojených na CAN sběrnici, kde CAN se nacházel již v mezních situacích. Lin je levná sběrnice vycházející z ISO 9141. Je využívána v oblasti, kde není nutná vysoká přenosová rychlost a spolehlivost, např. v ovládání a polohování zrcátek, stahování oken, ovládání zámek dveří a střešního okna, polohování sedadel, ovládání klimatizace, stěračů nebo osvětlení.

Čtvrtá kapitola je zaměřena na FlexRay sběrnici. Tato sběrnice je plně redundantní, deterministická a odolná proti poruchám systému. Pracuje s velmi vysokou přenosovou rychlostí a spolehlivostí. Je vyvíjena a implementována z důvodu pokročilých automobilových systému. FlexRay má sloužit pro takové aplikace automobilu, jako jsou brzdění a řízení tzn., že budou nahrazeny hydraulické systémy plně elektronickými.

V následující části práce je zpracováno K vedení a L vedení (ISO9141). Jedná se o diagnostické vedení, které je nejčastěji spojeno s bránou CAN sběrnice.

Popis jednotlivých vedení je soustředěn nejprve v obecné rovině s popisem historie vzniku a jejich základní vlastnosti. Dále rozebírá fyzickou vrstvu sběrnice, kde je pojednáno o

možnostech provedení fyzického média případně topologii a je popsána problematika elektromagnetického rušení. V kapitolách byla rovněž popsána problematika linkové vrstvy, tzn. rámců, filtrování zpráv, zabezpečení proti chybám atd. Poslední části hlavních kapitol je uvedena struktura jednotky na příslušné sběrnici.

V práci jsou rovněž zpracovány komunikační protokoly transportní a aplikační vrstvy koncernu WW.

V poslední části je realizace jednotky K – Line. V přílohách je pak schéma hardwaru, DSP a navržený laboratorní protokol, který je vypracován v deskách diplomové práce.

V přílohách jsou rovněž zmapovány integrované obvody pro realizaci jednotek jednotlivých vedení.

2 Sběrnice CAN (Controller Area Network)

Sběrnici CAN je definována, jako sériový komunikační datový protokol, který byl vyvinut firmou Robert Bosch počátkem 80. let pro využití v automobilech [5]. Snahou této kapitoly je detailně popsat vznik CAN sběrnice, vlastnosti, fyzickou vrstvu, linkovou vrstvu a nástin hardwarového řešení jednotky CAN sběrnice, včetně průzkumu trhu integrovaných obvodů pro realizaci CAN jednotky.

2.1 Historie vzniku CAN

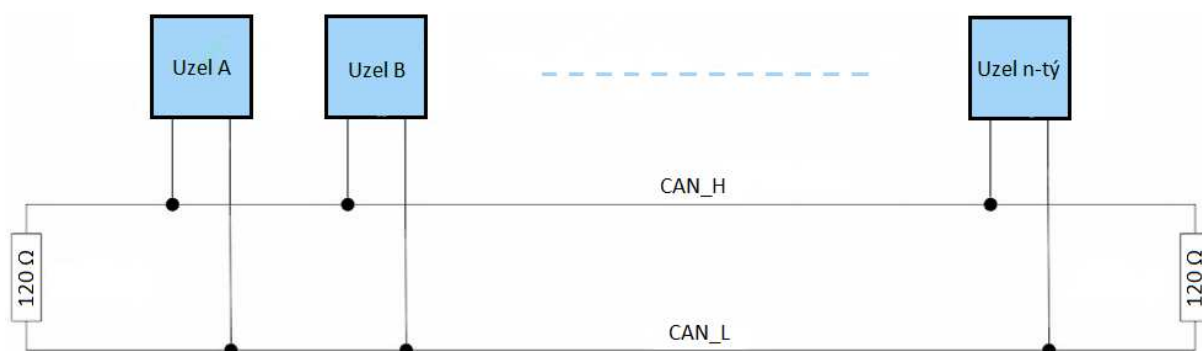
- 1983 Inženýři firmy Bosch zahájili projekt vývoje komunikační sítě pro motorová vozidla. Firma Mercedes – Benz začala zpracovávat specifikace nové sériové komunikační sběrnice. Prof. Dr. Wolfhard Lawrenz v Braunschweig-Wolfenbattel v Německu, byl hlavní konzultant veškerých prací nad novým protokolem a pojmenoval ho Controller Area Network. Prof. Dr. Horst Wettstein z Karlsruhe byl jeho asistent.
- 1986 Vzdána oficiální informace k CAN protokolu na kongresu SAE v Dittroit. Systém (leden) byl prezentován Boschem, jako Automotive Serial Controller Network.
- 1987 Firmy Philips a Intel uvedly první obvody pro CAN.
- 1991 Vydání publikace CAN 2.0 (specifikace).
- 1992 Firma Mercedes-Benz uvádí první automobil se sběrnici CAN.
- 1993 Představení standardu ISO11898
- 1995 Standart ISO 11898 je změněn a doplněn o rozšířený rámec.

2000 Vývoj time-triggered (časově-spouštěného) komunikačního protokolu pro CAN (TTCAN).

Zajímavost: Ve Škodě Auto (stará Octavia) byla sběrnice CAN použita od roku 1996 [3], také VW v tomto roce začal CAN sběrnici implementovat [5].

2.2 Základní vlastnosti CAN sběrnice

Vlastnosti CAN sběrnice splňují požadavky automobilových aplikací a stává se standardem v Evropě pro automobilové sběrnice [5]. Sběrnice CAN pracuje se sériovým přenosem dat a umožňuje distribuovaný řízený systém v reálném čase, její maximální přenosová rychlost je 1Mbit/s s vysokým stupněm zabezpečení dat. Data všech uzlů jsou přenášena po jednom vedení - sériově. Všechna zařízení mají k přenosovému médium přístup jako multi – maestr [1], čili každý uzel sběrnice může být jako hlavní a řídit takto jiný uzel. Rozhodování, který uzel na síti je hlavní (master) probíhá na základě nižší priority. Není potřeba tedy řídit celou síť z jednoho nadřazeného uzlu, takto je dosaženo vyšší spolehlivosti a zjednodušení. Ukázka CAN sítě je na obrázku 1, kde jsou znázorněny i odpory, které se musí vyskytovat u sítě tohoto typu (s ohledem na délku vedení), aby byly eliminovány odrazy na vedení.



Obrázek 1: principiálně struktura CAN sběrnice

Veškeré zprávy vysílané po sběrnici neobsahují adresu cíle, čili zprávu obdrží všechny uzly na sběrnici. Uzly poznají na základě identifikátoru, zda jim patří [6].

2.2.1 Princip komunikace

- ✓ Peer to peer – uzly jednoho systému jsou si rovné.
- ✓ Rámce – Jsou uzlem odesílány do sítě a přijímány všemi uzly najednou.
- ✓ U komunikace není adresace uzlů, rámec vždy obsahuje jedinečný identifikátor, který vypovídá o obsahu zprávy.
- ✓ Je-li zpráva úspěšně přijatá alespoň jednou jednotkou na síti, je vysílači potvrzeno doručení zprávy.
- ✓ Je-li detekována chyba libovolného komunikujícího uzlu, je vysílána chybová sekvence bitu a proces se opakuje ihned.
- ✓ Každý uzel zpětně přijímá to, co vysílá.

2.3 Parametry CAN sběrnice - obecně

Teoreticky na CAN síť můžeme připojit n-tý počet uzlů, ale norma ISO 11898 podle literatury [4] udává maximální počet 30 zařízení připojených k síti a to z důvodu, aby byly zajištěny statické a dynamické parametry sběrnice.

Zavedení parametrů CAN BUS		
<i>Přenosová rychlost</i>		maximálně 1 Mbit/s
<i>Počet komunikujících zařízení</i>		maximálně 30
<i>Standardní impedance</i>		120 Ω
Max. délka vedení	1 Mbit/s	40 m
	500 kbit/s	112 m
	300 kbit/s	200 m
	100 kbit/s	640 m
	50 kbit/s	1340 m
	20 kbit/s	2600 m
	10 kbit/s	5200 m

Tabulka 1: Parametry CAN sběrnice podle ISO 11898

Přehled přenosové rychlosti je uvedená v tabulce a však obecně lze říci pro všechna přenosová média, že čím delší vedení tím menší rychlost. Dále je třeba si uvědomit, že

přenosová rychlost je závislá na dalších skutečnostech, např. typ vedení, standard fyzického média.

2.4 Vrstvy CAN sběrnice

Specifikace CAN 2.0 z roku 1991 zavádí dvě základní vrstvy, které vycházejí z ISO/OSI modelu. Důvodem zavedení těchto vrstev je dosažení transparentnosti (průhlednosti) a flexibility implementování návrhu s touto sítí. Na obrázku 2 je zaznamenán ISO/OSI referenční model, přičemž CAN specifikace 2.0 zavádí šedě vyznačené vrstvy. Tedy v této fázi kapitoly nás zajímá objektová vrstva, transportní vrstva a fyzická vrstva.

1.	Aplikační vrstva
2.	Prezentační vrstva
3.	Relační vrstva
4.	Transportní vrstva
5.	Síťová vrstva
6.	Linková vrstva (CAN) <i>Objektová vrstva LLC (Logical Link Control)</i> <ul style="list-style-type: none"> ○ Filtrování přijatých zpráv, ○ Hlášení o přetížení, ○ Manipulace se zprávami a stavy <i>Transportní vrstva MAC (Medium Access Control)</i> <ul style="list-style-type: none"> ○ Detekce, kontrola a signalizace chyb, ○ Řízení rámců, ○ Potvrzování přijatých zpráv, ○ Přenosová rychlost a časování, ○ Řízení přístupu uzlů k přenosovému médium
7.	Fyzická vrstva (CAN) <ul style="list-style-type: none"> ○ Definuje úroveň signálu a reprezentaci jednotlivých bitů, ○ Definuje přenosové médium

Obrázek 2: Referenční model ISO/OSI

2.5 Fyzický vrstva (přenosové médium)

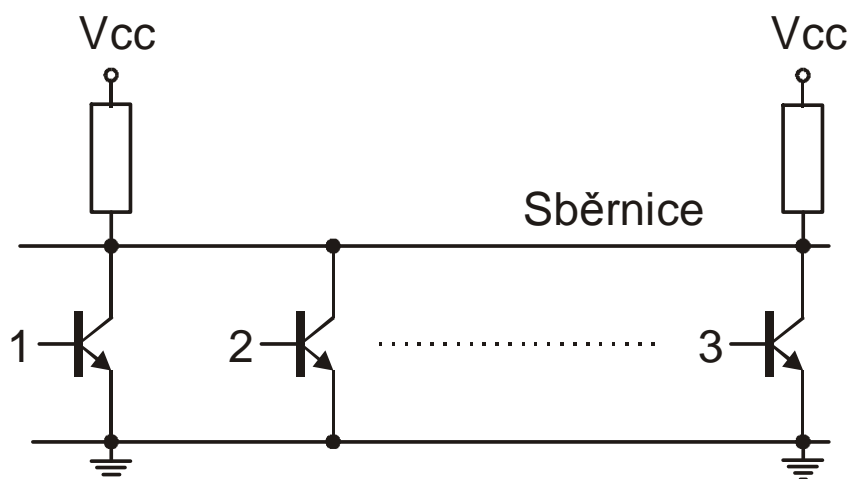
Zadává, jakým způsobem jsou data vysílána s ohledem na elektrické vlastnosti. Fyzická vrstva je stejná pro všechny komunikující zařízení v dané síti, přičemž je možné definovat parametry sítě pro konkrétní aplikace.

Fyzické přenosové médium protokolu CAN je navržen tak, aby realizoval funkci logického součinu. Standardem CAN jsou definovány dvě logické úrovně bitu – *dominant* (aktivní stav) a *recessive* (pasivní stav). Je třeba poznamenat, že stav *dominant* odpovídá logické 0 a *recessive* odpovídá stav logické 1. Logické hodnoty nemají striktně dány napěťové úrovně, resp. liší se v závislosti na použitém standardu ISO (viz. níže), zde je třeba ještě říci, že specifikace CAN protokolu nepojednává o typu fyzické vrstvy. Chování bitů na sběrnici je zcela jasné a vychází ze základu logických obvodů, čili vysílají li všechny uzly sběrnice bit *recessive* (logickou 1), je na síti *recessive* úroveň. Vyšle li alespoň jeden uzel bit *dominantní* (logická 0), bude na sběrnici *dominant* stav, znázorněno v tabulce 2.

V.b. 2. u.		Vysílaný bit 1. uzelem	
		Dominant	Recessive
	Dominant	Dominant	Dominant
	Recessive	Dominant	Recessive

Tabulka 2: Logický součin AND

Princip chování sběrnice je možné uvést na příkladu sběrnice buzené hradly s otevřeným kolektorem (obrázek 3). Je-li jeden z tranzistoru sepnutý, objeví se na sběrnici úroveň logické 0, tzn. *dominant*. V případě, že ani jeden tranzistor není sepnut na sběrnici je úroveň *recessive* (log 1).



Obrázek 3: Příklad fyzické vrstvy CAN protokolu

Pro realizaci fyzického média se většinou používá diferenciální (rozdílový) signál. V další části si popíšeme některé normy fyzické vrstvy s ohledem na automobilový průmysl a také rozeberme rušení na síti.

2.5.1 Normy fyzické vrstvy

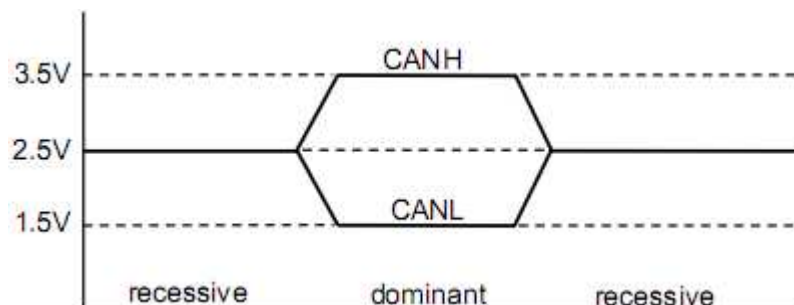
Normy definují například rozdílové napětí, druh vedení a tak podobně. V další části si uvedeme některé podstatné z norem ISO 11898 – 2, ISO 11898 – 3, ISO 11519, ISO 11992 pro fyzické médium.

Norma ISO 11898 – 2 (CAN High Speed)

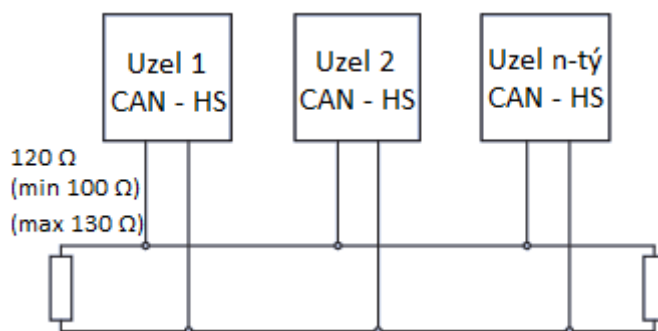
- ✓ Přenosová rychlost dat je 1Mbit/s při délce vedení 40m (detailně v tabulce 1)
- ✓ Typická je sběrniceová struktura se zakončovacími odpory
- ✓ Až 30 uzlů připojených k síti
- ✓ Nutnost ukončovacích odporů 120Ω na konci sítě
- ✓ Jednotky mohou být vzdáleny od sběrnice 0,3m
- ✓ Definován také jako standard SAE
- ✓ Diferenciální komunikace

Důležité: Pro dosažení kompatibility všechny uzly na síti musí obsahovat stejný nebo podobný bitová takt. Pro automobilové aplikace - SAE publikuje SAE J2284 specifikaci.

Poznámka: Délku, nebo počet uzlů CAN sběrnice lze zvýšit pomocí mostů (bridge), které tak propojují několik sítí.



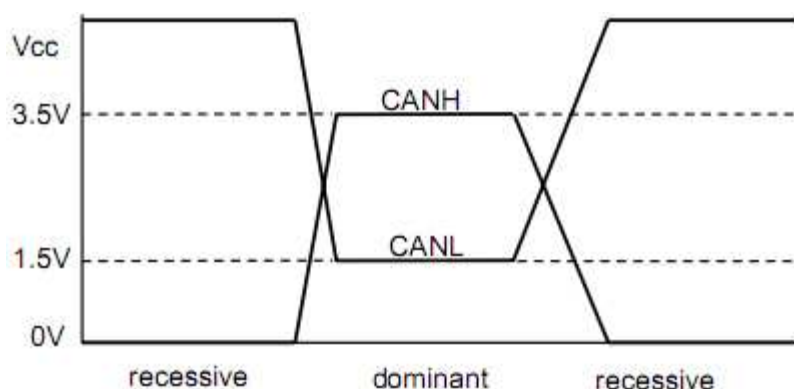
Obrázek 4: Pásmo napěťových úrovní logických stavů sběrnice CAN podle norma 11898 - 2



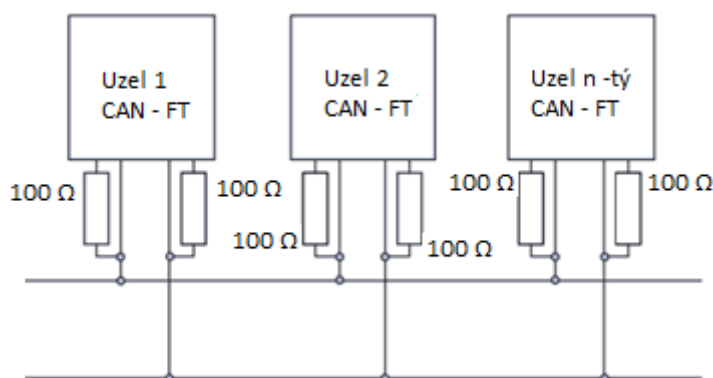
Obrázek 5: Propojení uzlů na sběrnici CAN podle normy 11898 – 2

Norma ISO 11898 – 3 (fault – tolerant)

- ✓ Přenosová rychlost dat je 125kbit/s
- ✓ Sběrníková struktura
- ✓ Až 32 uzlů (jednotek) na síti
- ✓ Napěťová úroveň se pohybuje od 0V do +5V
- ✓ Napájení je vymezeno pro 5V
- ✓ Zakončení sítě (terminátory) se odvíjí od počtu uzlů v síti
- ✓ Definován také jako standard SAE
- ✓ Tolerance k určitým chybám, je schopna přepnout na jednodrátový provoz (a to v případě zkratu jednoho z vodičů na kostru nebo mezi sebou).



Obrázek 6: Pásmo napěťových úrovní logických stavů sběrnice CAN podle norma 11898 - 3



Obrázek 7: Propojení uzlů na sběrnici CAN podle normy 11898 – 3

Norma ISO 11519 (Low – Speed)

- ✓ Nepotřebuje zakončovací rezistory (na síti nevznikají odrazy z důvodu nízké rychlosti)
- ✓ Definován také jako standard SAE

ISO 11519 Low - Speed		
Signál	Stav recessive	Stav dominant
CAN H	1.75V	4.0V
CAN L	3.25V	1.0V

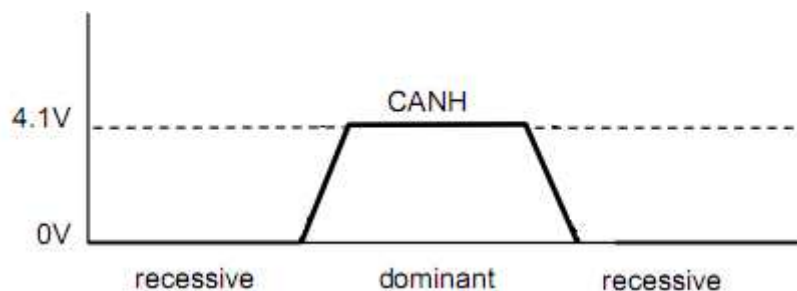
Tabulka 4: Napěťové úrovně logických stavů sběrnice CAN podle standardu ISO 11519

Norma SAE J2411 (Single wire)

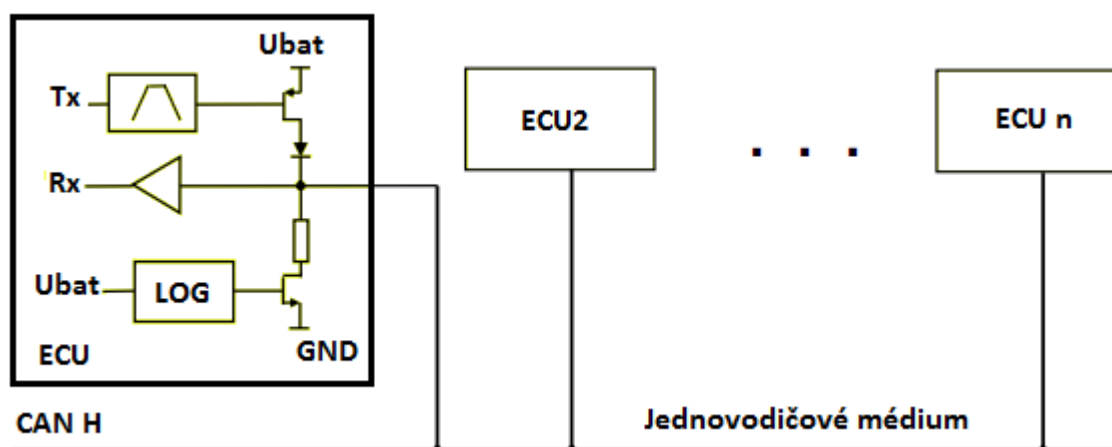
- ✓ Přenosová rychlost dat je 33,3 kbit/s
- ✓ Umožňuje vysokorychlostní přenosový mód pro diagnostiku (83,3kbit/s)
- ✓ Jedná se o jednovodičový standard CAN sběrnice (nestíněný jednovodič)

- ✓ Až 32 uzlů (jednotek) na síti
- ✓ Napěťová úroveň se pohybuje od 0V do +4,1V
- ✓ Není potřeba ukončovacího odporu
- ✓ Maximálně 40 m vedení

Aplikace: Aplikuje se v oblasti komfortní elektroniky motorových vozidel



Obrázek 8: Pásmo napěťových úrovní logických stavů sběrnice CAN podle norma SAE J2411



Obrázek 9: Propojení uzlů na sběrnici CAN podle normy SAE J2411

Norma ISO 11992

- ✓ Přenosová rychlost 125kbit/s (40m).
- ✓ Napájecí napětí 12V nebo 24V

Aplikace: Dvoubodové spojení pro použití tažných vozidel a přívěsu.

Dílčí závěr

Jsou zde uvedeny standardy, které jsou používány v automobilovém průmyslu, přičemž poslední uvedený standard patří do skupiny nákladních automobilů. Existují i další standardy popisující fyzické médium v zemědělských strojích, medicíně a tak dále. Tímto problémem se zabývá CAN in Automation založena šéfredaktorem VMEbus Holger Zeltwangerem.

2.5.2 Zamezení rušení na sběrnici CAN

Striktní dodržování délky vedení včetně přenosové rychlosti, dále hodnoty terminátoru atd., již je jistý předpoklad pro nerušený přenos. Tyto jistě vynikající parametry jsou uvedeny výše, v této podkapitole se zaměříme na fyzikální provedení přenosového média.

Aby se snížilo rušení vnějšími elektromagnetickými poli (např. magnetické pole alternátoru) jsou oba vodiče zkrouceny (obrázek 10). Oba vodiče vedou stejný signál, jen opačné hodnoty (obrázek 11). CAN sběrnice tedy vede diferenciální signál a je jasné z teorie o významu kroucených vodičů, že okamžitý součet napětí na obou vodičích je konstantní a případná rušení jsou vzájemně eliminována. Při takovém principu vedení je parazitní indukce, kapacita a impedance relativně malá:

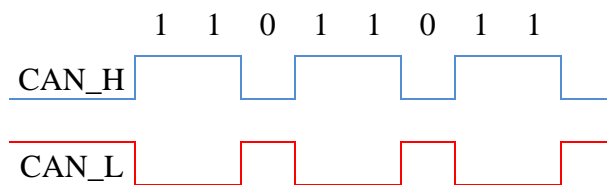
$$L = 0,4\mu \left(2,3 \log \frac{2A}{d} + 0,25 \right), \quad C = 12,1 \frac{\varepsilon_{ef}}{\log \left(\frac{2A}{d} \right)}, \quad Z = \frac{276}{\sqrt{\varepsilon}} \log \frac{2A}{d}, \quad \left[\frac{\mu H}{m} \right], \left[\frac{pF}{m} \right], [\Omega].$$

[9], (1), (2), (3)

kde ε_{ef} je efektivní relativní permitivita dielektrika, přibližně $\varepsilon_{ef} = 0,8\varepsilon$, μ je relativní permeabilita.



Obrázek 10: Rozměrové parametry krouceného dvojitého vodiče [9]



Obrázek 11: Uspořádání bitů na vedení

Poznámka: V automobilové technice se používá vedení např. FL6Y2G 296

2.6 Linková vrstva

Podobně jako v ISO/OSI modelu i zde linkovou vrstvu dělíme na dvě pod vrstvy:

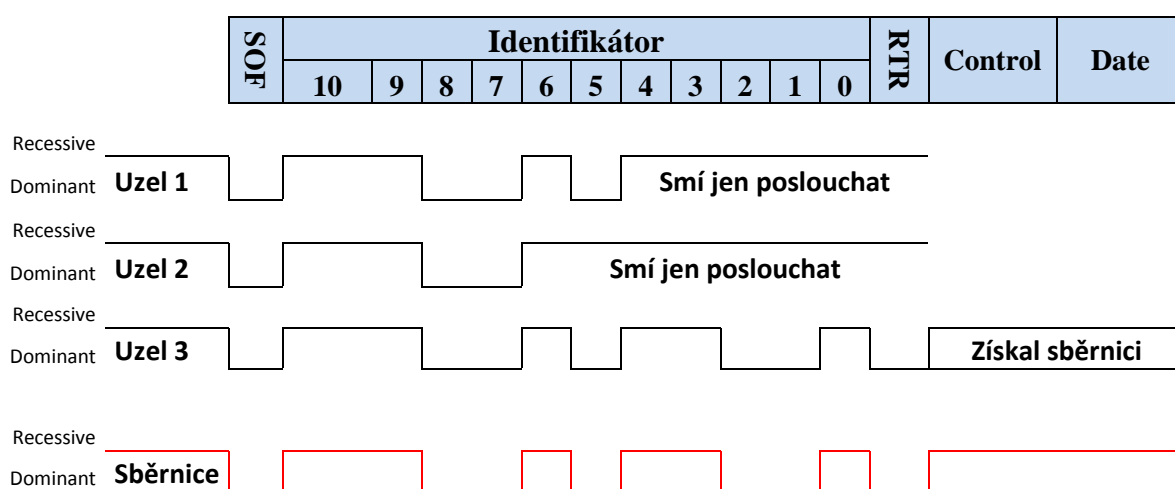
- ✓ MAC (*Medium Access Control*) – obecně řečeno tato vrstva zabezpečuje přístup k médiu, tzn., že kóduje data, doplňuje je o doplňkový bit, řídí přístup všech jednotek (dle priority), detekuje a hlášení chyb a udává informaci o bezchybně přijatých zprávách.
- ✓ LLC (*Logical Link Control*) – obecně řečeno řídí datové spoje tzn., že provádí filtrování zpráv (*Acceptance Filtering*) a hlášení o přetížení sítě (*Overload Notification*).

2.6.1 Řízení přístupu k médiu

Vysílá-li současně několik uzlů téže sítě ve stejný čas, pak sběrnice je přístupna tomu, který vysílá zprávu s vyšší prioritou. Vyšší prioritu má ta zpráva, která má nižší identifikátor. Každý vysílající uzel porovnává hodnotu právě vysílajícího bitu s hodnotou na sběrnici a na základě pravidel priority zastaví vysílání (vysílající uzel vysílá *recessive* bit a na sběrnici se právě vyskytuje *dominant* úroveň), nebo začne vysílat (vysílající uzel vysílá *dominantní* bit a na sběrnici je *recessive* úroveň). Čili, tak je zabezpečeno, že zpráva s nižším identifikátorem (vyšší prioritou) může vysílat přednostně. Situaci znázorňuje obrázek 12. Je-li sběrnice volná, může libovolný uzel vysílat. Po přenosu rámce mají jiné uzly přístup k sběrnici. Pro identifikátor obecně platí:

- ✓ Musí být jedinečný (dva různé uzly nesmí vysílat rámeček se stejným identifikátorem)
- ✓ Má šířku 11 bitů (rozšířený 29)
- ✓ Je vysílán na počátku zprávy (hned po bitu SOF (*Star of Frame*), který je vždy dominantní)
- ✓ Je odesílán od nejvyššího bitu (obvykle se liší v nejnižším bitu)
- ✓ Každý uzel přijímá rámce s těmi identifikátory, které ho zajímají.

Poznámka: Vyskytne-li se chyba v právě přenášené zprávě, odesílá se chyba hned po zjištění chyby.



Obrázek 12: Předávání přístupu ke sběrnici při komunikaci na CAN Bus

2.6.2 Časování a synchronizace bitů

Je zcela jasné, že všechny uzly v síti musí mít shodnou nominální rychlost, ovšem skutečná rychlost se elementárně liší, což je způsobeno oscilátory. Dále je třeba vzít v potaz, že více uzlů může vysílat najednou a že šíření informace z jedné jednotky do druhé je zatíženo zpožděním, tzn. budič – vedení – přijímač. Z těchto důvodů je synchronizace nutností. Děje se tomu tak, že programovatelná dělička generuje signál s délkou nazvanou časové kvantum. A dle obrázku 13 je pak vidět, že z uceleného počtu kvant je složen bitový interval.

Tolerance oscilátoru lze vypočítat dle dvou požadavků, první požadavek odpovídá startovacímu bitu a druhý je počítán s ohledem na bitový tok):

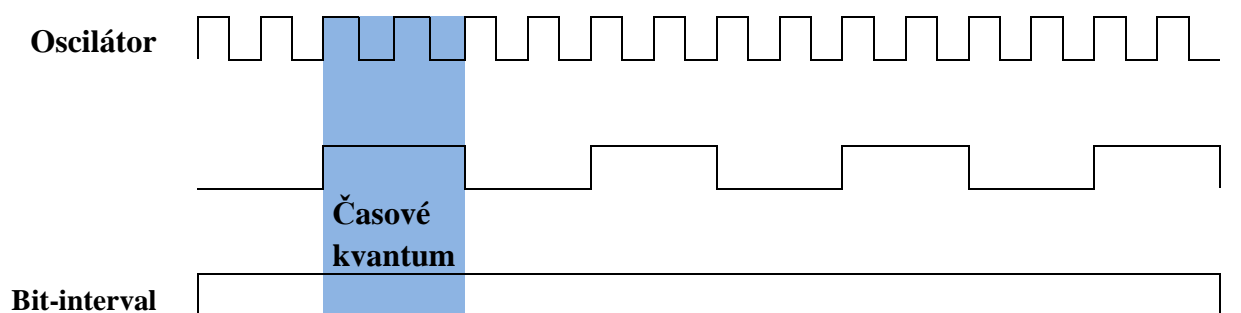
$$dF < \frac{\min(t_{P_SEG1}, t_{P_SEG2})}{2 \times (13 \times t_{BitTime} - t_{P_SEG2})}, \quad dF < \frac{t_{SJW}}{2 \times (10 \times t_{BitTime})}$$

[8], (4), (5)

Příklad výpočtu: Rychlost = 125 Kbps, časový interval = 16tq, synch_segment = 1tq, prop_segment = 3tq, fázový segment 1 = 6tq, fázový segment 2 = 4tq, šířka synchronizačního skoku (SJW) = tq

První požadavek $dF < 1,48\%$

Druhý požadavek $dF < 1,25\%$

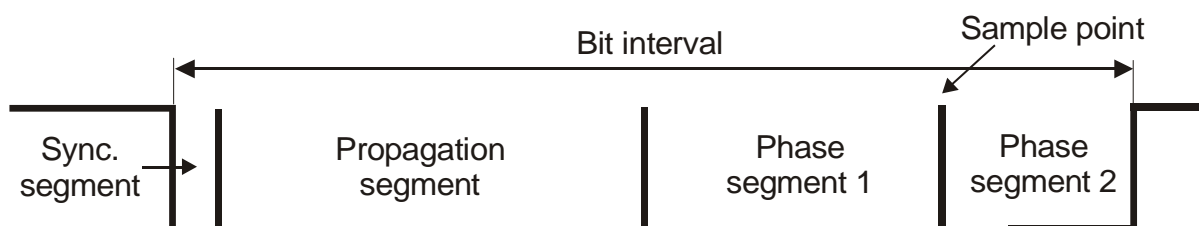


Obrázek 13: Princip časových kvant

Z výše uvedeného je jasná dělba bitového intervalu a už víme, že na začátku každé zprávy probíhá synchronizace přenosu, tzv. tvrdá synchronizace. Děje se tak pomocí sestupné hrany SOF bitu. Synchronizační proces probíhá rovněž při každé změně logické úrovně recessive na dominantní úroveň, pak je stávající časové kvantum bráno za synchronizační segment. Bitový interval se skládá z 8 až 25 časových kvant, které tvoří čtyři segmenty, situace je znázorněná na obrázku 14:

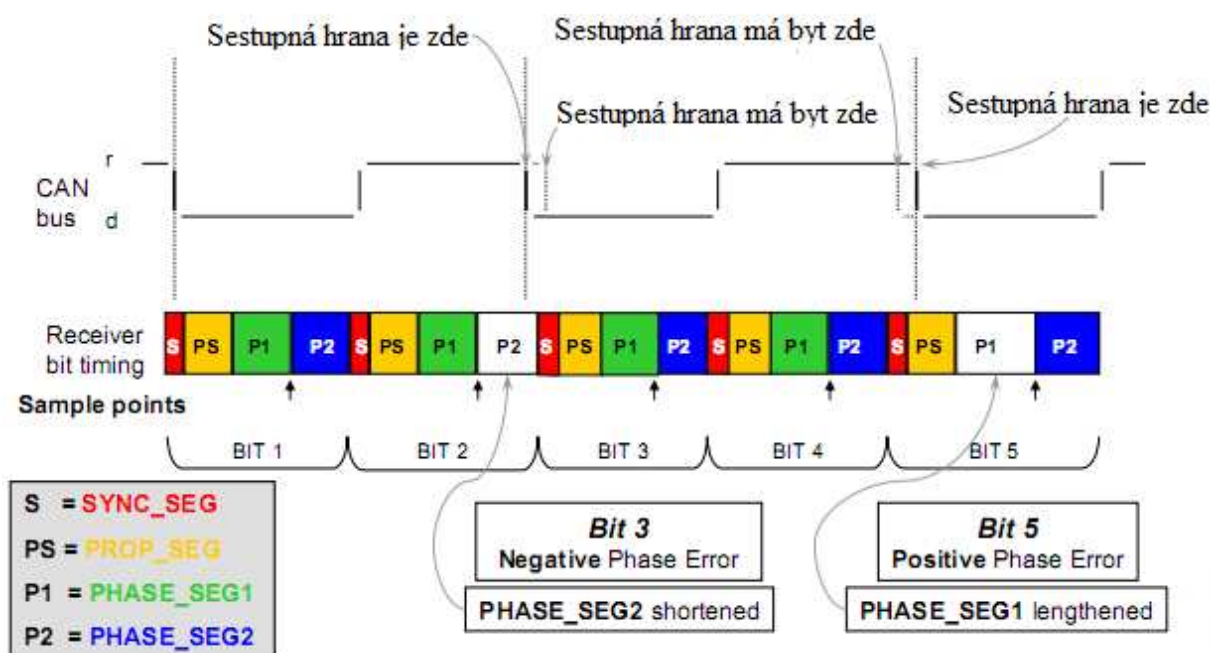
- ✓ Synchronization Segment – první synchronizační úsek pro synchronizaci na hranu signálu. Je dlouhý 1 časové kvantum.
- ✓ Propagation Time Segment – kompenzace signálových zpoždění mezi uzly, zpoždění vzniká vlivem samotného šíření signálu a zpožděním převodníků u CAN uzlů. Je dlouhý 1 až 8 časových kvant.
- ✓ Phase Segment 1 – kompenzace fázových poruch hran (segment se může prodloužit). Je dlouhý 1 až 8 časových kvant.
- ✓ Sample point – doba během, které se sejme ze sběrnice úroveň.

- ✓ Phase Segment 2 – kompenzace fázových poruch hran (segment se může zkrátit).



Obrázek 14. Struktura bitu na CAN sběrnici

Pro lepší pochopení principu synchronizace, uveďme si příklad. Při kladné fázové chybě je fázový segment 1 (P1) prodloužený tak, aby byl bod vzorkování správně zpožděný za příchozí hranou signálu, resp. můžeme říct, že dojde k zpoždění taktu. A při záporné fázové chybě je fázový segment 2 (P2) zkrácen tak, aby příslušná hrana signálu se nacházela v synchronizačním segmentu následujícího bitu. Toto slovní přiblížení je patrné i z obrázku 15.



Obrázek 15. Znázornění řešení taktové chyby [8]

2.6.3 Zabezpečení přenášených dat a detekce chyb

Protokol CAN disponuje několika mechanismy pro zabezpečení dat. Tyto mechanismy pracují současně. Obecně lze napsat, že mechanismus detekce chyb CAN sběrnice má tyto schopnosti:

- ✓ Globálně detekuje všechny chyby
- ✓ Detekuje lokální chyby ve vysílačích
- ✓ Odhaluje v rámci přerušení menší než 15 bit
- ✓ zjistí výskyt jakéhokoliv nepatřičný bit ve zprávě

Zajímavost: Pravděpodobnost neodhalení chyby při přenosu dat je menší, než:

$$\text{počet chybných zpráv} * 4,7 * 10^{-11}$$

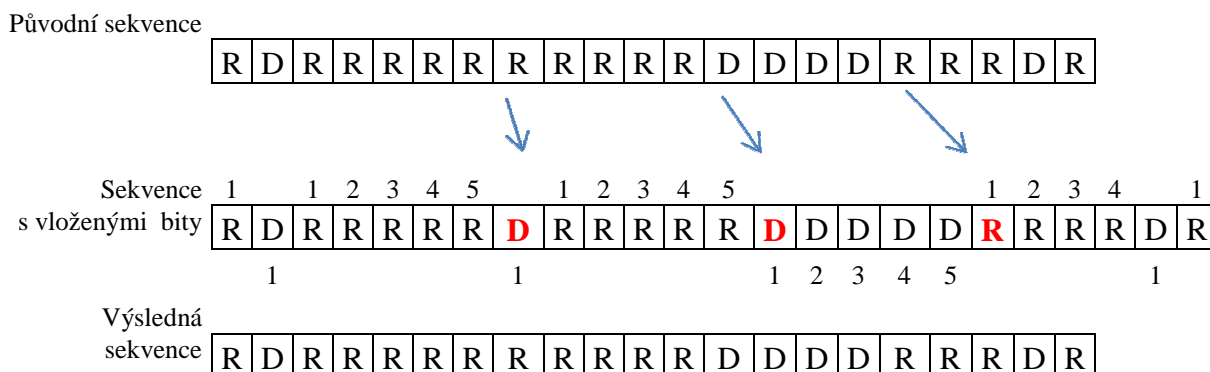
(6), [7]

Monitoring – Vysílač porovnává hodnotu vysílaného bitu s úrovní hodnoty na sběrnici (tzn., že zpětně přijímá vysílané bity). Mohou nastat dva stavy:

1. Detekuje-li v poli arbitráže *dominant* stav sběrnice a vysílá *recessive* bit, tak přestane vysílat
2. Detekuje-li v poli arbitráže *recessive* stav sběrnice a zatím co vysílá dominantní bit, nebo detekuje-li kdekoli jinde (krom ACK) jiný stav sběrnice, než je hodnota právě vysílaná, tak vyšle chybový rámec

CRC zabezpečení – Zde jsou přenášena data zabezpečena 15 bitovým CRC (*Cyclic redundancy check*) kódem. Jedná se o cyklický redundantní součet, který je generován ze všech předcházejících bitů zprávy, podle polynomu $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$. Je-li detekována chyba CRC libovolnou jednotkou na sběrnici, je vygenerována chyba CRC.

Vkládání bitu (*bit stuffing*) – Je-li vysílána sekvence pěti (nebo více) bitů shodné hodnoty, je vždy po pěti stejných bitech vložen jeden bit opačné úrovně. Po pěti přijatých bitech stejné hodnoty musí mít následující bit opačnou logickou úroveň a ví se, že se jedna o vložený bit a vyjme se. Situace je zachycená na obrázku 16. Tato kontrola se děje jednak kvůli kontrole dat, ale také zabezpečuje správné časové sesynchronizování přijímačů jednotlivých jednotek.



Obrázek 16: Znáznornění vkládání bitu

Kontrola formátu zprávy (*Message Frame Check*) – Zpráva se kontroluje podle dané specifikace (některé bity mají definovanou hodnotu – konec rámce, oddělovač CRC, potvrzení) tzn., že pokud je zjištěná na nějaké pozici bitu zprávy nepovolená hodnota, generuje se chyba formátu rámce.

Potvrzení přijetí zprávy (*Acknowledge*) – Příjme-li jednotka zprávu v pořádku, přijetí se potvrdí změnou úrovně jednoho bitu zprávy (ACK). ACK bit je vždy vysílán s úrovní *recessive* a detekuje-li vysílač *dominant* hodnotu, tak je vše v pořádku. Potvrzení musí vyslat všechny jednotky na síti a to bez ohledu zda je filtrování zpráv zapnuto, či ne. Vysílání se opakuje, pokud není detekován ACK bit.

2.6.4 Chybové stavy uzlů (signalizace chyb)

Problémy s vysíláním (přijímáním) jedné jednotky by mohlo zcela zastavit komunikaci. V řadiči CAN sběrnice a tedy i v každém uzlu sítě jsou implementovány tzv. chybové čítače (jeden pro vysílání a jeden pro příjem). Čítače jsou na začátku nulovány. Při bezchybném přenosu je příslušný čítač dekrementován. Uzel se může nacházet v jednom ze tří chybových stavů:

1. Aktivní (*Error Active*) – Některými literárními zdroji je nazýván varovný příznak (*Error Warning Bit*). Jednotky se podílejí aktivně na komunikaci, a pokud detekují chybu vyšlou šestici po sobě jdoucích *dominant* bitů, nastal tzv. aktivní příznak (*Active Error Flag*). Pro zachování aktivního stavu, každý s chybových čítačů nesmí mít hodnotu větší nebo rovnou 127.

2. Pasivní (*Error Passive*) – Jednotky se podílejí na komunikaci s tím, že vysílají pouze pasivní příznak chyby (*Passive Error Flag*), který je definován, jako šestice po sobě jdoucích *recessive* bitů – dojde k destrukci zprávy. Tento stav nastane tehdy, když jeden z čítačů inkrementoval přes 127. Po snížení hodnot pod 127 obou čítačů jednotky, se smí vrátit do Aktivního stavu. Pokud hodnota chybového čítače překročí inkrementační mez 255, přejde do stavu – odpojeno.
3. Odpojený (*Bus-off*) – Pokud uzel generuje více než 255 chyb, je logicky odpojen a nemá přístup k médiu (nelze vysílat rámce, přijímat je teoreticky možné). Tento stav je definitivní a vzniká, když nastanou „fatální chyby“ (např. zkrat obou linek na kostře), opustit ho lze jen programově – resetem řadiče.

V běžném správném provozu jsou všechny uzly v aktivním stavu a chybové zprávy se v komunikaci nevyskytují. A jednotka, která je v síti jediná, přejde po 16 nepotvrzených zprávách do pasivního stavu a v něm zůstane.

2.6.5 Filtrování zpráv

Úkolem linkové vrstvy je filtrování zpráv. Fyzikálně se tato činnost děje v řadiči uzlu. Existují dvě principy filtrování zpráv:

- ✓ BasicCAN – Je variantou hardwarově jednodušší a tedy i levnější. Tato varianta filtrování obsahuje dva registry. Selektor identifikátoru přejímané zprávy, kde jsou nastaveny hodnoty odpovídající jednotlivým bitům identifikátoru. Aby však identifikátor nebyl zaměřen pouze jen na jeden identifikátor, lze nastavit masku, která vynechá určité bity identifikátoru, resp. je ignoruje a tak se uzel zpřístupní pro více zpráv.

Řadiče BasicCAN mají standardně dva přijímací bufery a jeden vysílací. V případě zpracování více zpráv, ukládají se do vyrovnávací paměti (bufery) typu FIFO. Zprávy jsou zpracovávány z jedné paměti FIFO, zatím co do druhé již mohou být ukládány. V případě zaplnění obou vstupních pamětí jsou zachovány informace nejstarší, z toho plyne, že jestliže procesor nestíhá informace zpracovávat s dostatečnou rychlostí, tak zprávy mohou být ztraceny.

Zpráva je vysílána přes zápis do vysílací paměti řadiče.

- ✓ FullCAN – Je variantou hardwarově dražší a je používán v rychlejších řadičích, resp. v rychlejších procesorech. Má pouze selektor. Maska odpadá.

Řadič FullCAN obsahuje paměti, tzv. *mailboxes*.

Když řadič přijímá informaci, musí ověřit ve svých mailboxech, jestli má takový mailbox se stejným identifikátorem, jako příchozí informace. Pokud je nalezen takový mailbox, tak informace zůstane zachována a je informován procesor.

Když procesor vysílá zprávu, tak je zapsána délka dat a samotná data do vysílajícího mailboxu s odpovídajícím identifikátorem.

Každý řadič FullCAN má omezený počet mailboxes. Pokud je stávající mailbox zaplněn, tak starší informace je ztracena (u BasicCAN je starší informace zachována).

2.7 Formát rámce CAN protokolu

Specifikace CAN protokolu zavádí čtyři typy rámců:

- ✓ Datový rámec
 - určený k přenosu dat
 - proměnná délka (0 – 8 bajtů dat)
- ✓ Rámec žádost o data
 - žádá o vyslání datového rámce s tímž identifikátorem
 - neobsahuje data
- ✓ Chybový rámec
 - šest po sobě jdoucích stejných bitů (1 nebo 0)
 - vysílá se, když je detekovaná chyba
- ✓ Rámec přetížení
 - stejná bitová posloupnost, jako chybový rámec
 - Vysílá se z důvodů odložení vysílání

První dva uvedené rámce se týkají komunikace jako takové a poslední dva uvedené rámce slouží k managementu komunikace (signalizace detekovaných chyb, vyžádání prodlevy v komunikaci, vyloučení chybových zpráv).

2.7.1 Formát datového rámce

CAN Specifikace definuje dvě varianty, které se od sebe liší počtem bitů v identifikátoru:

- ✓ CAN 2.0A – je kompatibilní s druhou variantou a využívá 11 bitový identifikátor.
- ✓ CAN 2.0B – představuje dva typy rámců a to standardní a rozšířený.

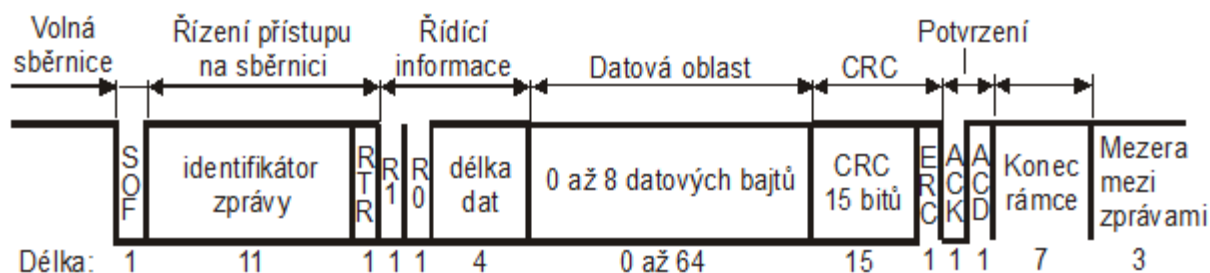
Oba standardy se mohou vyskytovat v jedné síti a to za předpokladu použití řadiče podporující 2.0B standardu.

Datový rámec podle standardu 2.0A

Standardně datový rámec se skládá ze 7 úseku.

- ✓ Počáteční pole SOF (*Start of Frame*) je jednobitové a má vždy hodnotu dominant (logická úroveň 0).
- ✓ Pole arbitráže (*Arbitration Field*) se skládá z identifikátoru (11 bit) a RTR bitu. RTR (*Remote Transmission Request*) je jednobitový a pokud je nastaven na úroveň logické 0 (*dominant*), tak se jedná o datovou zprávu. V opačném případě se jedná o žádost o data.
- ✓ Řídící pole (*Control Field*) obsahuje jednak rezervované bity R0, R1 a dále obsahuje 4 bity určující počet přenášených bajt ($0 \div 8$).
- ✓ Datové pole (*Data Field*) je tvořeno datovými bajty zprávy (maximálně 8 bajtů).
- ✓ CRC pole (*CRC Field*) obsahuje 15 bitový CRC kód a ERC *recessive* bit, jako CRC oddělovač.
- ✓ Potvrzující pole (*ACK Field*) je tvořeno ACK potvrzením a ACD oddělovačem potvrzení (*recessive*).
- ✓ Pole konce rámce (*End Of Frame*) je dáno 7 *recessive* bity

Dále nastupuje 3 bitová mezera (*Intermission field*), která je tvořená *recessive* bity. Po této bitové sekvenci je sběrnice volná libovolný čas s logickou nulou.

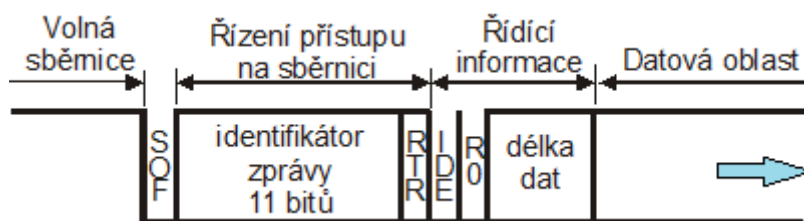


Obrázek 17: Datová zpráva dle specifikace CAN 2.0A

Datový rámec podle standardu 2.0B

Specifikace 2.0B vznikla, aby bylo dosaženo větší kompatibility s jinými protokoly sériové sběrnice v automobilových aplikacích v USA [2] a umožňuje vyšší počet rámců v systému. Z tohoto důvodu je tato specifikace zpětně kompatibilní se specifikací CAN 2.0A. Struktura rámce specifikace 2.0B se liší od předešlé varianty tím, že obsahuje dvě arbitrážní pole. První arbitrážní pole je 11 bitů dlouhé pro splnění kompatibility a druhé pole čítá 18bit.

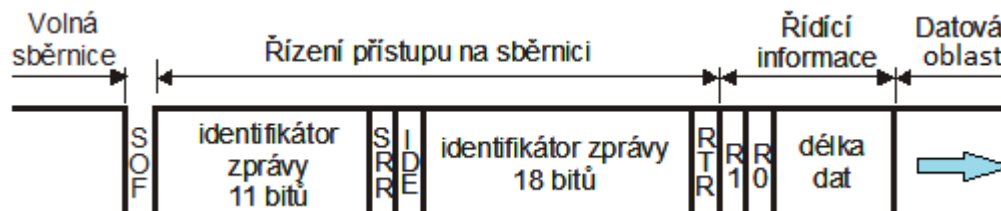
V zásadě specifikace 2.0B hovoří o standardním a rozšířeném formátu. Standardní formát (*Standard Frame*) rámce je vidět na obrázku 18 a jediným rozdílem od standardu CAN 2.0A je využití rezervovaného bitu R1 na indikaci zda se jedná o rámec standardní nebo rozšířený. Tento bit se podle specifikace CAN 2.0B označuje IDE (*Identifier Extended*). Je *dominantní* pro standardní rámec a *recessive* pro rámec rozšířený.



Obrázek 18: Datová zpráva dle specifikace CAN 2.0B (standardní rámec)

Rozšířený formát rámce (*Extended Frame*) používá 29 bitů pro identifikaci. Struktura je jasná z obrázku 19, přičemž je zaměněn bit RTR na bit SRR (*Substitute Remote Request*), který má vždy hodnotu *recessive*. Toto řešení je potřebné při kolizi rozšířeného a standardního rámce, aby přednost dostal standardní rámec. Zbylé bity odpovídají výše popsanému. Ostatní části, jsou rovněž stejné, jak ve specifikaci CAN 2.0A.

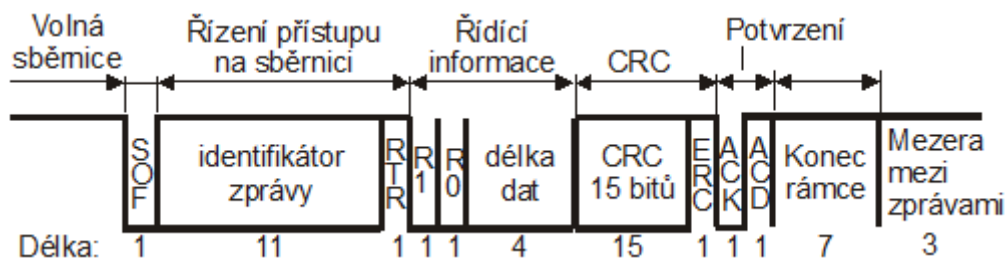
Poznámka: Úroveň priority jednotlivých identifikátorů je specifikací CAN 2.0B dána v následujícím pořadí ID (11bit), SRR, IDE, ID (18bit), RTR.



Obrázek 19: Datová zpráva dle specifikace CAN 2.0B (rozšířený rámec)

2.7.2 Formát rámce žádosti o data podle CAN 2.0A (B)

Rámec žádosti o data je podobný datovému rámcu. Tedy z výše uvedeného je jasné, že bit RTR musí být *recessive*. Identifikátor rámce je shodný s identifikátorem zprávy, o jehož obsah žádáme. Tím je zajištěno to, že pokud ve stejný čas jeden uzel žádá o data a druhý mající stejný identifikátor vysílá data, přednost získá uzel vysílající datovou zprávu (hodnota RTR datového rámce vysílajícího uzlu má úroveň *dominant* – čili má vyšší prioritu. Pole délka dat je vždy 0, tedy rámec žádosti o data neobsahuje datové pole (viz obrázek 20).

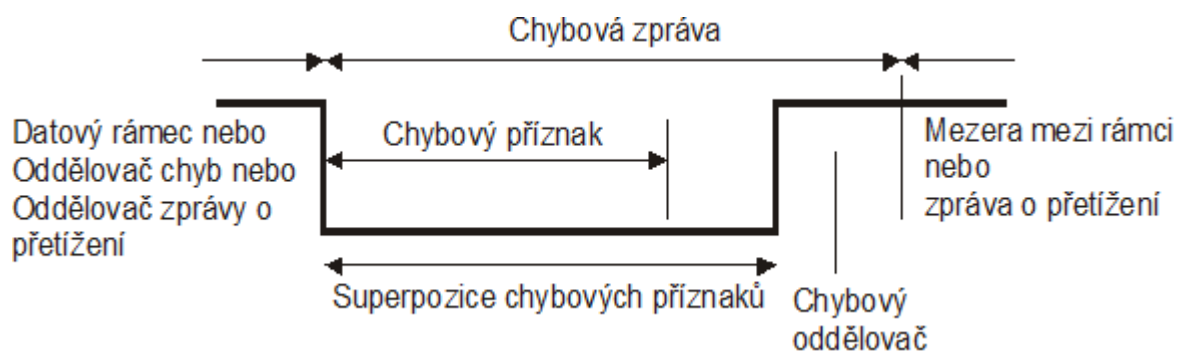


Obrázek 20: Zpráva žádosti o data dle specifikace CAN 2.0A

Poznámka: Pro specifikaci CAN 2.0B platí analogie k popsanému principu CAN 2.0A.

2.7.3 Formát chybového rámce

Chybová zpráva vypovídá o vzniku chyby na některém z uzlu. Může se jednat o chybu CRC kódu, chyba bitu, chyba vkládání bitu nebo chybu rámce. Po detekování některé z uvedených chyb, patřičný uzel generuje chybový rámec s aktivním příznakem chyby (6 bitů *dominant*) nebo pasivním příznakem chyby (6 bitů *recessive*). Jestli se jedná o aktivní nebo pasivní příznak chyby, je to dáno tím, jestli vysílající uzel je v aktivním nebo pasivním signalizačním módu (viz kapitola 2.6.4). Při výskytu aktivního příznaku chyby je zpráva poškozena s ohledem na pravidla vkládání bitů. Při takové chybě vysílají i ostatní uzly na médiu chybové rámce, tímto vznikne superpozice chybových příznaků. Při detekování pasivního příznaku chyby, uzel vyšle 6 *recessive* bitů, které ale neovlivní stav sběrnice.



Obrázek 21: Chybový rámec

Chybový rámec se skládá z chybového příznaku (*Error Flag*), který je 6 bitový (s bity stejné úrovně) a chybového oddělovače (*Error Delimiter*), který je 8 bitový nízké úrovně. Chybový oddělovač vytváří místo pro ostatní uzly, které chtějí také vysílat vlastní chybové rámce (viz obrázek 21).

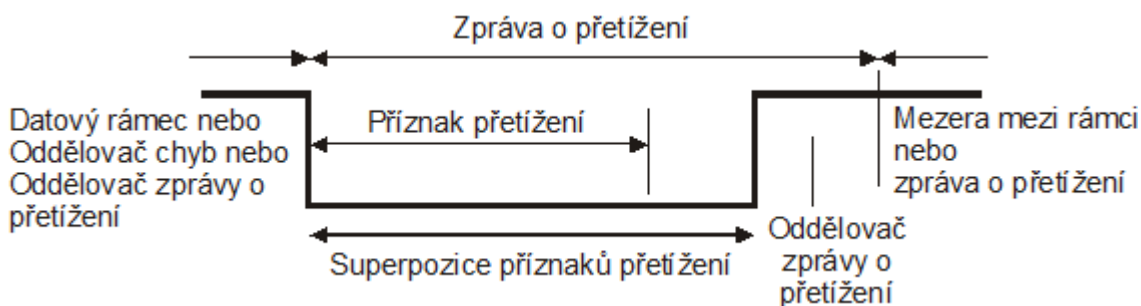
Poznámka: O vše se stará řadič, nikoliv aplikační software.

2.7.4 Formát rámce o přetížení

Zpráva o přetížení se skládá z příznaku o přetížení (6 *dominantních* znaků) a případné superpozice všech příznaků přetížení, za předpokladu generování přetížení více uzly.

Podobně, jak v chybové správě následuje sedm *recessive* bitů, které podle obrázku 22 tvoří oddělovač zprávy o přetížení. Zpráva o přetížení se liší od chybové zprávy tím, že zpráva o přetížení může být vyslána, až na konci předcházejícího rámce, nikoliv ihned. Dále neznamená chybný příjem rámce a nedochází k opakování předchozí zprávy.

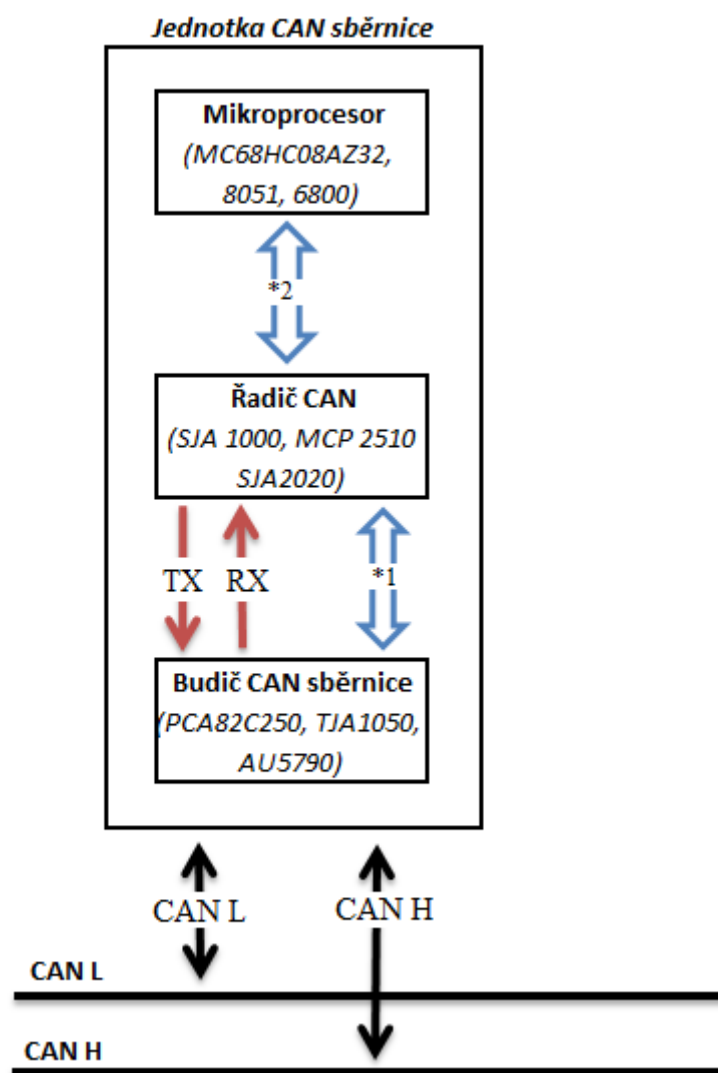
Poznámka: Vysílá ho přijímací uzel. Dnešní řadiče jej nepoužívají z důvodu pozdržení příjmu.



Obrázek 22: Rámec přetížení

2.8 Hardwarové řešení jednotky CAN sběrnice

Podoba hardwarového terminálu ve funkci řídicí jednotky na CAN sběrnici je blokově znázorněna na obrázku 23.



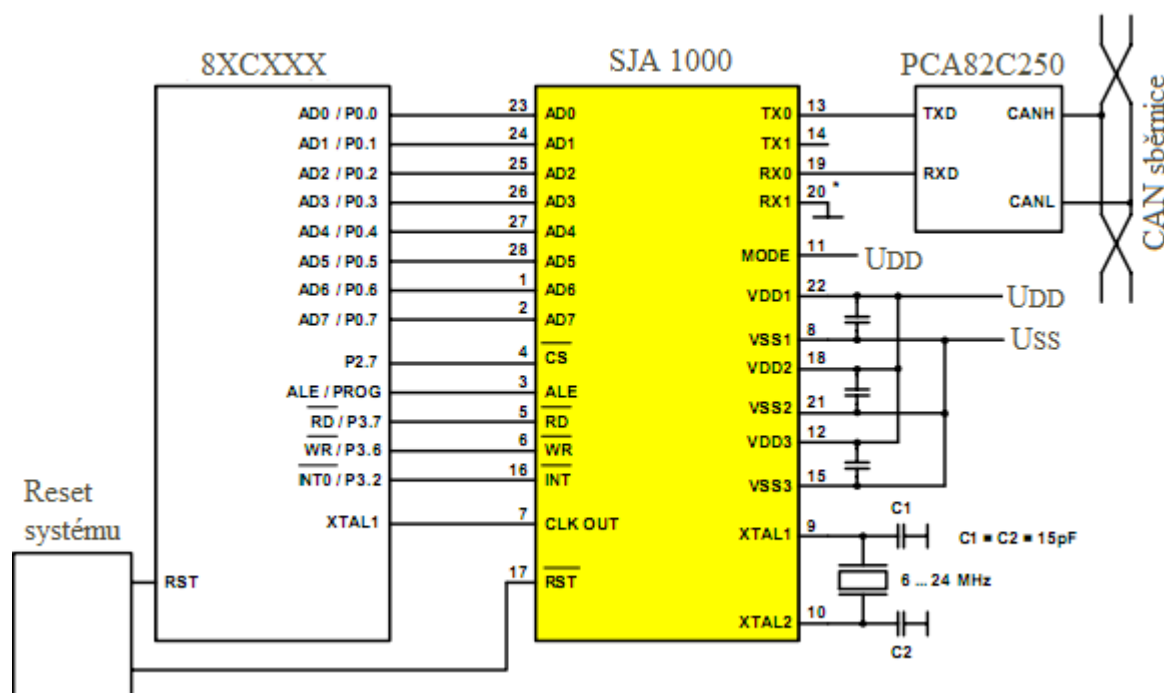
(*1 řídicí signál, *2 paralelní nebo sériová sběrnice)

Obrázek 23: Typické zapojení jednotky na CAN Bus

- ✓ Mikroprocesor - Obsahuje události, zpracovává, vysílá a přijímá data na sběrnici CAN.
- ✓ Řadič CAN (CAN Controller) – Zpracovává datovou linkovou vrstvu protokolu CAN, tj. má na starosti rámce, zabezpečení dat, prioritu a filtrování zpráv.
- ✓ Budič CAN (CAN transceiver) – Realizuje fyzickou vrstvu protokolu CAN. Má tedy za úkol převést TTL úroveň (od řadiče) na úroveň fyzické vrstvy CAN sběrnice (na diferenciální signál patřičné úrovně, podle použité normy).

Poznámka: Konkrétní přehled integrovaných obvodů pro realizaci CAN jednotky je v příloze 3.

Na dalším obrázku je uvede doporučené schéma hardwarového terminálu CAN sběrnice.



Obrázek 24: Zapojení CAN jednotky [18]

Funkce obvodu je jasná z uvedeného v kapitole 2.8, přičemž některé literární zdroje [19] zapojení doplňují o DC optočleny.

V aplikačních poznámkách [18] je k dispozici materiál, který popisuje programování procesoru pro účely inicializace a tak dále.

3 LIN sběrnice (Local Interconnect Network)

Stávající metody, kdy se data přenášejí přes jednotlivá kabelová spojení se nacházejí v mezní situaci. Proto se zvyšujícím nasazením řídicích jednotek zapojených na sběrnici CAN začíná také prosazovat sběrnice LIN.

Dalším důvod prosazení této sběrnice je, že od samého počátku byla navrhována s ohledem na cenu. Literatura [12] uvádí, že náklady jsou 2 – 3 krát nižší ve prospěch Lin. Sběrnice LIN nenahrazuje CAN vedení, pouze ji doplňuje vhodným způsobem, tedy není třeba takových vysokých přenosových rychlostí a vysoká bezpečnost přenosu. Dále pro konstrukci není třeba používat speciálních řadičů, postačí běžný mikropočítač s obvodem UART. Také postačí jednoduchý RC členek ve funkci krystalu a tak dále.

Pro svůj charakter konstrukce je používána zejména v ovládání a polohování zrcátek, stahování oken, ovládání zámků dveří a střešního okna, polohování sedadel, ovládání klimatizace, stěračů nebo osvětlení.

Zajímavost: Do Octavia II je LIN Bus implementována do 3 subsystému [25]:

- ✓ Centrální řídicí jednotka vozu
- ✓ Centrální řídicí jednotka komfortní elektroniky
- ✓ řídicí elektronika sloupku řízení

3.1 Historie LIN sběrnice

- | | |
|--------------------|---|
| 1998
(říjen) | Vzniká pracovní skupina pracující na nové automobilové sběrnici |
| 1999
(červenec) | První specifikace 1.0 sběrnice s názvem LIN. Byla vytvořena zdržením LIN, která se skládala ze sedmi výrobců automobilového průmyslu a elektroniky: Audi, BMW, DaimlerChrysler, Motorola, VCT, VW, Volvo. |

2000	Prezentováno veřejnosti
2000 (duben)	Specifikace 1.1 LIN sběrnice
2000 (listopad)	Specifikace 1.2 LIN sběrnice
2001	První nasazení sběrnice LIN v automobilech
2002 (listopad)	Specifikace 1.3 LIN sběrnice
2003 (říjen)	Specifikace 2.0 LIN sběrnice (provedené významné opravy)
2006 (listopad)	Specifikace 2.1 LIN sběrnice, k datu vzniku této DP poslední specifikace. (úprava konfigurace, rozšíření transportní vrstva a přidání diagnostické prvky)

3.2 Základní vlastnosti LIN sběrnice

V podstatě jde o sběrnici typu single – master / multiple – slave. Z uvedeného vyplývá, že řídicí zařízení kontroluje komunikaci s jedním nebo s více podřízenými jednotkami. Dalším charakteristickou vlastností je jedno vodičové provedení s možností napojení se na síť pomocí logické funkce AND. Jedná se o sériový přenos dat formátu UART. Maximální přípustná rychlost přenosu dat pro toto vedení je 20kbit/s (2400 – 19200 bit/s). Pro usnadnění implementace jsou však doporučeny tyto rychlosti 2400 bit/s - Slow, 9600 bit/s - Medium a 19200 - Fast bit/s s délkou dat 2, 4 nebo 8 bajtů. Synchronizaci komunikace provádí master zařízení na počátku každého přenosu. Mezi další parametry je možno uvést:

- ✓ Typické impedanční zakončení pro nadřízenou jednotku 1 k Ω a pro podřízenou jednotku 30 k Ω
- ✓ Doporučený počet uzlů jednoho subsystému 17 (16 podřízených jednotek + 1 nadřízená jednotka)
- ✓ Maximální délka vodiče je 40m

- ✓ Komunikace je založená na UART / SCI
- ✓ Rychlost přeběhu je $2\text{V}/\mu\text{s}$
- ✓ Zakončovací kapacita je $220\text{ pF} / 2,2\text{ nF}$ (master / slave)
- ✓ Kapacita sběrnice je $100 \div 150\text{ pF/m}$

3.3 Princip komunikace

Hlavní jednotka typu Master obsahuje *master task* a *slave task*. Master ovládá veškerou činnost na sběrnici. Jedná se zejména o detekci chyb, synchronizaci, přenosovou rychlost, nízko příkonový mód (*sleep mode*). Master rovněž obsahuje přesný oscilátor, protože je zdroj synchronizačního signálu pro celou síť. *Master task*, umístěný v uzlu masteru, vysílá hlavičky rámců podle plánovací tabulky. Plánovací tabulka obsahuje pořadí identifikátorů rámců a interval prodlevy od ukončení daného rámce do začátku dalšího. Aplikace v master může obsahovat více těchto tabulek a může podle potřeby mezi nimi přepínat.

Podřízená jednotka typu Slave obsahuje pouze *slave task*. *Slave task* je zodpovědný za čtení nebo zápis dat do něj podle daného identifikátoru a dále detekuje hlavičku. Z důvodů neustále synchronizaci hodin s hodinami Master jednotky u slave uzlu postačí RC článek.

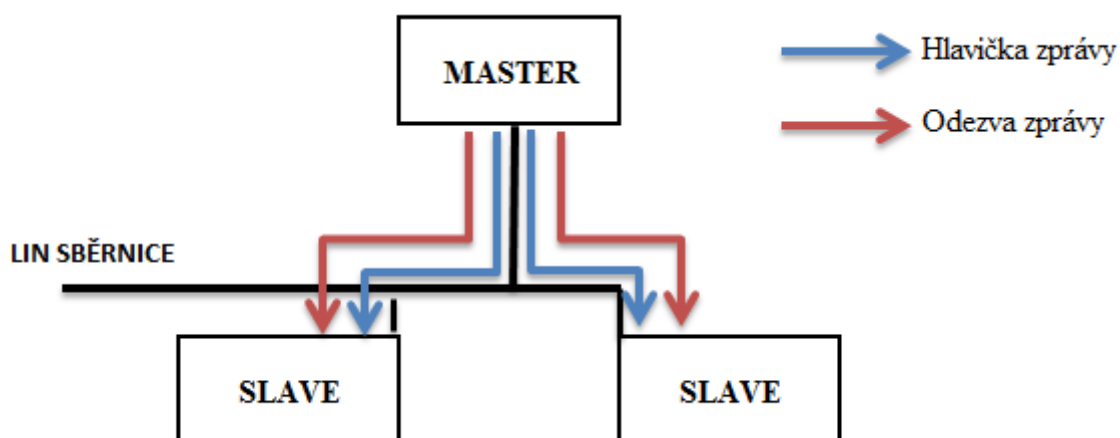


Obrázek 25: Základní uspořádání sběrnice LIN

3.3.1 Směry komunikace

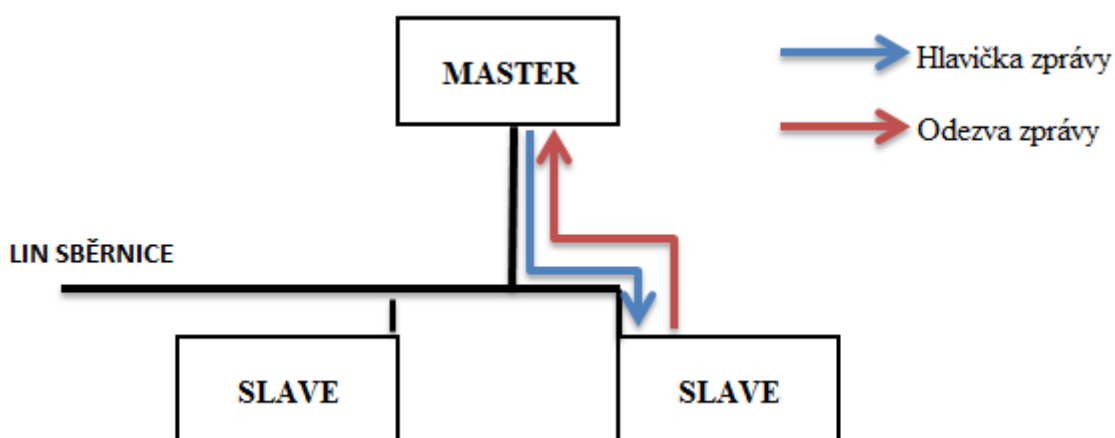
Komunikace po LIN sběrnici má 3 možné směry komunikace vycházející z koncepce sítě.

$Master \xrightarrow{směr} Slave$ – Nejčastěji se jedná o ovládní akčních členů (vypínání / zapínání motorků atd.). Master vyšle dva rámce, tedy příkazový rámec (hlavička zprávy) i Datový rámec (odezva zprávy) a to buď jedné jednotce, nebo více jednotkám. Situace je zobrazena na obrázku 26.



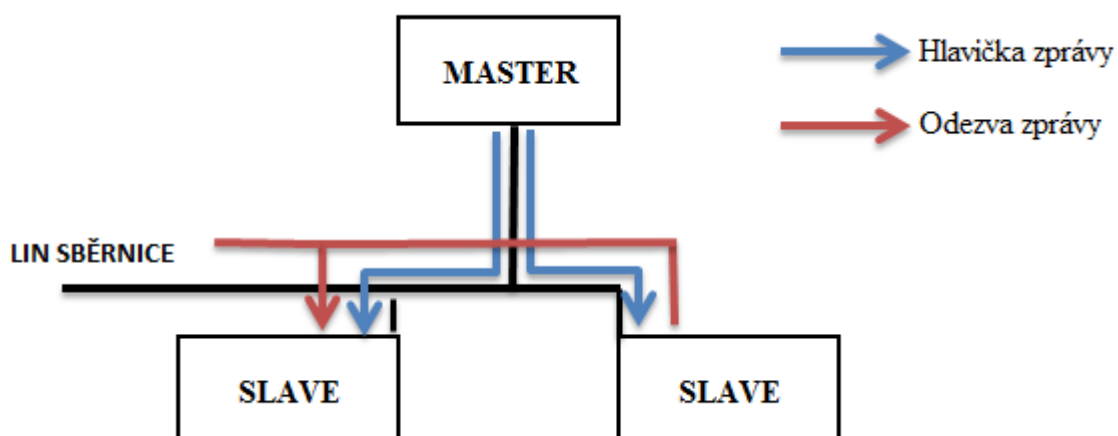
Obrázek 26: $Komunikace\ Master \xrightarrow{směr} Slave$

$Slave \xrightarrow{směr} Master$ – Nejčastěji se jedná o čtení stavových informací z podřízené jednotky. Tento požadavek na síti nastane tehdy, když master jednotka vyžaduje odpověď od určité podřízené jednotky. Tyto data nadřazený uzel zpracuje a provede např. nějakou změnu vstupně - výstupních veličin. Situace je zobrazena na obrázku 27.



Obrázek 27: $Komunikace\ Slave \xrightarrow{směr} Master$

$Slave \xrightarrow{\text{směr}} Slave$ – Tato možnost komunikace nastává tehdy, když není třeba data přenášet přes master jednotku. A podle obrázku 28 je vidět, že master jednotka vyšle příkazový rámec (hlavičku), na který odpoví slave jednotka vysláním dat, které však master nepotřebuje a použije je jiný slave.



Obrázek 28: Komunikace $Slave \xrightarrow{\text{směr}} Slave$

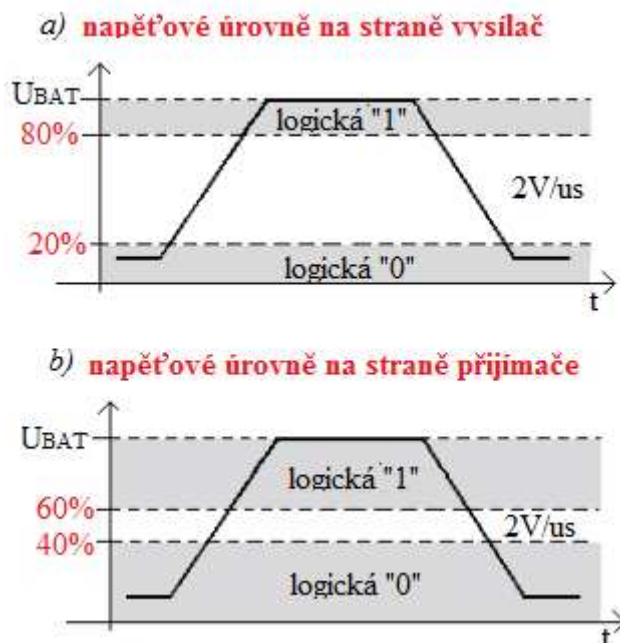
3.4 Fyzická vrstva

Fyzická vrstva LIN sběrnice je odvozená a od diagnostického standardu ISO 9141 (standard je popsán v práci). Oproti zmíněnému standardu je LIN vedení vylepšeno v oblasti rušení, kde je omezená strmost náběžných a sestupných hran. Rovněž logické hodnoty jsou definovány širším rozpětím napětí. Fyzická vrstva LIN vedení řeší i posun zemního potenciálu a různé poruchy.

Jak je popsáno výše sběrnice LIN je obousměrnou komunikaci po jednom vodiči pomocí realizace logické funkce AND (součinu) a rezistoru, které obsahuje každá jednotka v síti. Jak u ISO 9141, také zde jsou definovány dvě vzájemně komplementární hodnoty stavů – *dominant* (logická 0) a *recessive* (logická 1). Velikost úrovně napětí pro jednotlivé stavy jsou odvozena z napětí baterie automobilu vůči kostře, kde platí (obrázek 29)

- ✓ Logické „0“ odpovídá menšímu napětí než 20% z napětí baterie pro vysílač a pro přijímač menšímu než 40% z napětí baterie.

- ✓ Logická „1“ odpovídá většímu napětí než 80% z napětí baterie pro vysílač a pro přijímač většímu než 60% z napětí baterie.

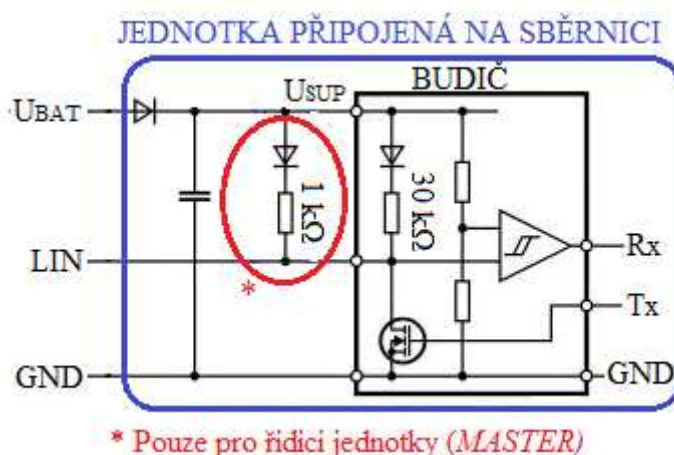


Obrázek 29: a) Napětíové úrovně pro přijímač

b) Napětíové úrovně pro vysílač

Pokud na sběrnici je alespoň jeden spínač (AND) sepnut (spojí vodič se zemí) sběrnice přejde do stavu *dominant*. Když není žádný spínač sepnutý, tak rezistory, které jsou zapojené mezi napětí baterie a sběrnici, na ní udržují stav logické jedničky (*recessive*).

V případě ztráty napájecího napětí musí být přítomný rezistory (zapojené do série s ochrannou diodou) určující stav *recessive*, aby byla zamezena nedefinovaná úroveň. Velikost rezistorů je definována jiná pro podřazené uzly i nadřazené uzly, přičemž nadřazený uzel má interní rezistor $30\text{ k}\Omega$ i externí (mimo budič) $1\text{ k}\Omega$, tento rezistor je nutný, aby se s vyšším počtem podřízených jednotek razantně neměnila velikost výsledného odporu, připojující sběrnici na napájecí napětí. Z uvedeného vyplývá, že budiče pro podřízenou a řídicí jednotku jsou stejné, liší se jinými externími součástkami. Další parametry fyzické vrstvy jsou uvedeny v kapitole „Základní vlastnosti LIN sběrnice“.



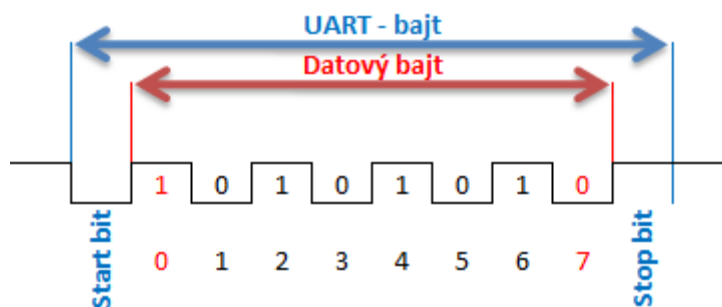
Obrázek 30: Zjednodušené schéma budiče v jednotce připojené na LIN sběrnici

3.4.1 Zamezení rušení na sběrnici LIN

Pro zabezpečení vyšší odolnosti proti elektromagnetickému rušení se paralelně k vývodu LIN budiče připojuje kondenzátor. Pro ještě lepší zamezení rušení, někteří výrobci používají RC filtry. Dalším významným vylepšením oproti ISO 9141 v oblasti spolehlivosti je rychlost přeběhu 2ms a rozšíření rozsahu pro jednotlivé logické úrovně.

3.5 Linková vrstva

U LIN protokolu jsou zprávy kódovány po bajtech, to vyplývá z implementování na UART protokolu (*Universal Asynchronous Receiver Transmitter*). Liší se pouze v synchronizační pauze (*synchronization break*).



Obrázek 31: Formát UART protokolu

Poznámka: Master řídí přenos na sběrnici a poskytuje přesnou synchronizaci celé sítě.

3.5.1 Formát rámce sběrnice LIN

Komunikace probíhá prostřednictvím rámce LIN zprávy (*LIN Message Frame*). Tento rámec se v zásadě skládá z 2 částí (obrázek 32):

- ✓ Příkazový rámec (*Header frame = Command frame*)
- ✓ Datový rámec (*Response frame = Data frame*)

Poznámka: Literatura [13] uvádí obě varianty anglických názvu.

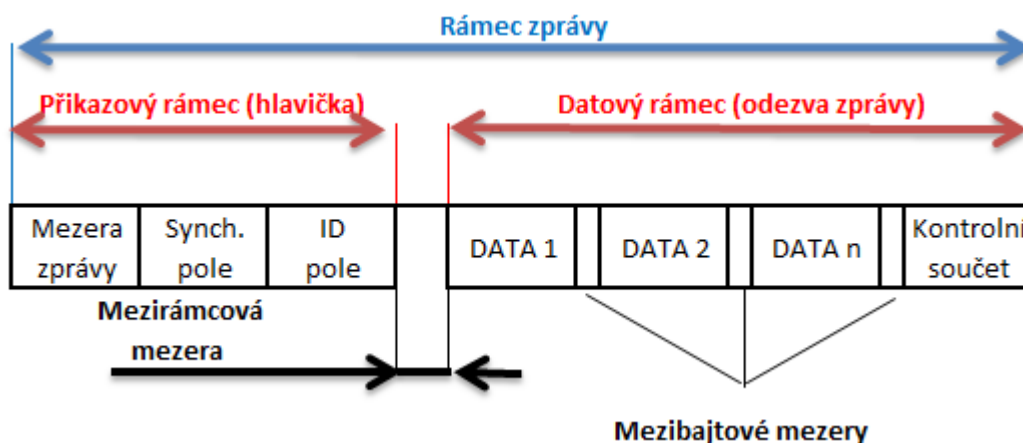


Obrázek 32: Rámec LIN zprávy

Příkazový rámec a datový rámec jsou odděleny mezirámcovou mezerou (*Interframe gab – IG*). Mezirámcová mezera má libovolnou délku a je definovaná, jako čas od příjmu žádosti slavem od mastru po vyslání odpovědi [14].

Příkazový rámec (resp. hlavička zprávy) se skládá ze tří polí a to z pole mezera zprávy, synchronizační pole a pole identifikátoru (obrázek 33). Příkazový rámec je odeslán z nadřazené jednotky. Datový rámec (resp. odezva zprávy) obsahuje 8 datových polí a pole kontrolního součtu. Datový rámec je odeslán z *slave tasku*.

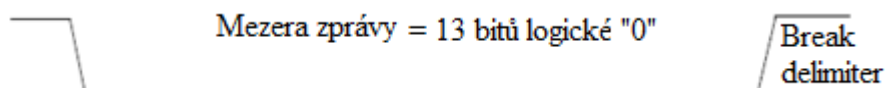
Komunikace je započata vysláním zprávy z nadřazeného uzlu Master (*master tasku*) do všech podřízených uzlů - *slave tasku*. Pomocí identifikátoru uzel rozpozná, zda a je zpráva určena pro něj. Následuje výměna zpráv mezi *slave tasky*.



Obrázek 33: Struktura rámce zprávy protokolu LIN

3.5.2 Mezera zprávy – synchronizační pauza

Je složena minimálně z 13 nulových bitů ukončených opačnou polaritou (tzv. *Break delimiter*). Tato pauza je zavedená z důvodu, aby všechny podřízené jednotky s dostatečnou spolehlivostí detekovaly zprávy vyskytující se na sběrnici. Podle specifikace 1.3 mohou mít podřízené jednotky vlastní hodinovou frekvenci odlišnou od hlavní jednotky, až o 15%, proto tak velká mezera zprávy. Pokud je detekována podřízenou jednotkou mezera zprávy, softwarová rutina musí zjistit, zda všechny přijaté bity jsou nulové. Mezera zprávy je znázorněná na obrázku 34.



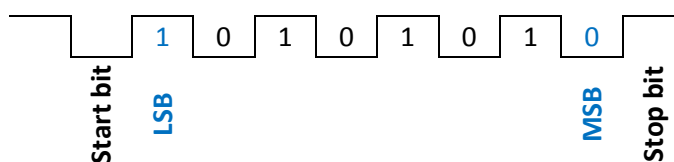
Obrázek 34: Pole mezery dat

Poznámka: V master jednotce, která je vystavěná na UART je problém udržet 13 nulových bitů po sobě, řešením podle literatury [13] je snížení přenosové rychlosti při generování

mezery zprávy. A tak např. 9 bitů generovaných v UART (při vhodné baud rate) nadřazené jednotce je v podřízené jednotce detekováno jako 13 nul.

3.5.3 Synchronizační pole

Nadřazená (Master) jednotka inicializuje přenos vysláním hlavičky, podle které podřízené jednotky synchronizují svoje hodiny před tím, než je přijata nová zpráva (tzn. Master slouží jako časová reference). Synchronizační pole je tvořeno 10 bity (obrázek 35), přičemž první bit představuje start bit (dominant), další bity (8bitů) jsou představeny střídáním se úrovní bitů a poslední bit je stop bit, který je na sběrnici definován recessive stavem. Hexa hodnota synchronizačního pole je 0x55.



Obrázek 35: Synchronizační pole

Synchronizace podřízených jednotek je dána odměřením času od první sestupné hrany (start bitu) k páté sestupné hraně synchronizačního pole (7bit) a následně je vyděleno 8, aby byla zajištěna přenosová rychlost mastru [15].

3.5.4 Pole identifikátoru

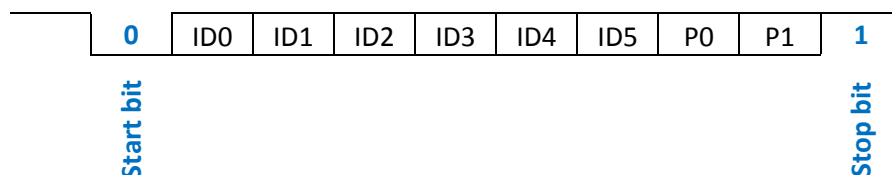
Oblast chráněného identifikátoru (*PID – Protected Identifier Field*) se nachází na posledním místě hlavičky zprávy. Identifikátor (6 bitů) nese informaci o odesilateli, příjemci a délce datové zprávy. Jedná se o chráněné pole, a tedy obsahuje zabezpečení pomocí dvou paritních bitů (6 a 7 bit, obrázek 36). Jejich hodnoty jsou definovány funkcí XOR významných bitů podle vzorce

$$P_0 = ID_0 \oplus ID_1 \oplus ID_2 \oplus ID_4$$

(7), [17]

$$P_1 = \overline{ID_1 \oplus ID_3 \oplus ID_4 \oplus ID_5}$$

(8), [17]



Obrázek 36: Pole identifikátoru

Podle uvedeného je zřejmé, že můžeme rozlišit kombinaci 6 bitů, tedy 64 různých identifikátoru. Podle použití je můžeme rozdělit do dílčích kategorií [17]:

- ✓ 0x00 až 0x3B (0 - 59) – Pro obecná data
- ✓ 0x3C až 0x3D (60 – 61) – Pro diagnostická data
- ✓ 0x3E (62) – Pro rámec uživatelem definovaný (rezerva)
- ✓ 0x3F (63) – Pro budoucí použití (rezerva)

Následuje datový rámec, které je variabilní a může se skládat z různého počtu bajtů, podle tabulky (Obrázek 37).

ID5	ID4	Počet datových bajtů
0	0	2
0	1	2
1	0	4
1	1	8

Obrázek 37: Délka datového pole

3.5.5 Datový rámec

Na LIN sběrnici v jednom datovém rámci se může vyskytovat až 8 datových polí. Datové pole (obrázek 38) se skládá z deseti bitů, přičemž se zde vyskytuje start bit (*dominant* úroveň), stop bit (*recessive* úroveň) a osm vlastních dat určených k přenosu.



Obrázek 38: Datové pole

3.5.6 Pole kontrolního součtu

Kontrolní součet (obrázek 39) je umístěn v rámci odezva. Starší specifikace sběrnice LIN do specifikace 1.3 požívali zabezpečení pouze datového pole. Od specifikace 2.0 se zavádí rozšířený kontrolní součet, který kontroluje datové pole i pole identifikátoru [16]. Použití standardního nebo rozšířeného kontrolního pole součtu určují nadřazený uzel v závislosti na použitém identifikátoru. Například identifikátory rámce 0x3C a 0x3D vždy budou používat klasický součet.



Obrázek 39: Pole kontrolního součtu

V tabulce 5 je ukázka zabezpečení dat pomocí kontrolního součtu. Pro zabezpečený přenos jsou zvolena čtyři datová pole. Vypočtený kontrolní součet je pak 0x26, dalším negováním výsledku dostaneme 0xD9. Přičemž součet kontrolního součtu a negovaného kontrolního bajtu musí vyjít 0xFF, jinak je přenos chybný. Příjímácí uzel může snadno kontrolovat shodu dat, použitím stejného početního mechanismu.

OPERACE		BIT DÁLE	D7	D6	D5	D4	D3	D2	D1	D0
1. datové pole	0x48 hex		0	1	0	0	1	0	0	0
2. datové pole	0x5E hex		0	1	0	1	1	1	1	0
3. datové pole	0x93 hex		1	0	0	1	0	0	1	1
4. datové pole	0xE5 hex		1	1	1	0	0	1	0	1
0x48+0x5E	0xA6		1	0	1	0	0	1	1	0
Přenos součtu	0xA6		1	0	1	0	0	1	1	0
0xA6+0x93	0x139	1	0	0	1	1	1	0	0	1
Přenos součtu	0x40		0	1	0	0	0	0	0	0
0x40+0xE5	0x125	1	0	0	1	0	0	1	0	1
Přenos součtu	0x26		0	0	1	0	0	1	1	0
Pole kontrolního součtu	0x26		0	0	1	0	0	1	1	0
Negace kontrolního součtu	0xD9		1	1	0	1	1	0	0	1
Pak platí 0x26+0xD9	0xFF		1	1	1	1	1	1	1	1

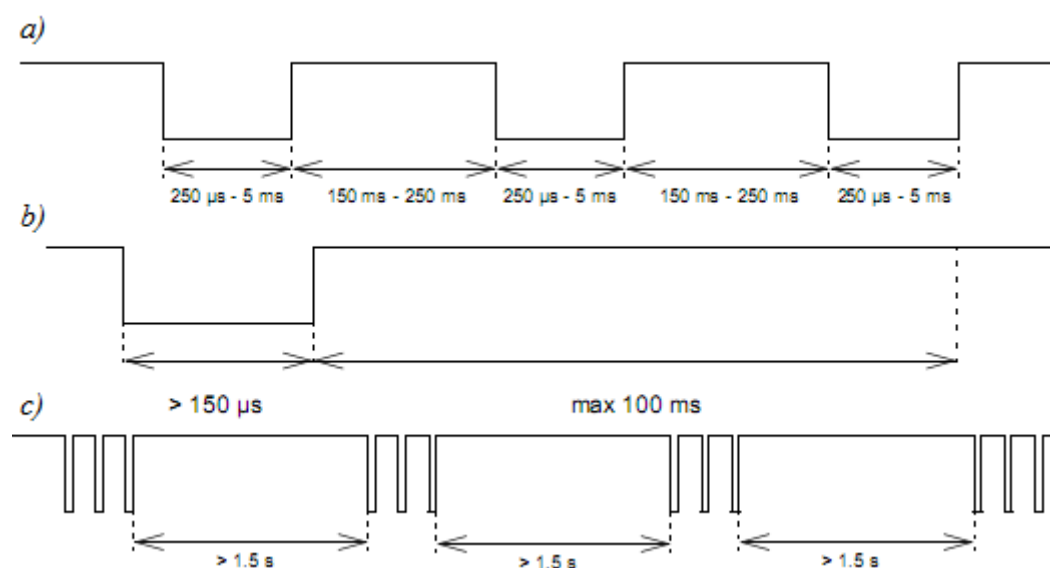
Tabulka 5: Ukázka kontrolního součtu čtyř datových polí [14]

3.6 Některé další funkce

Specifikace LIN sběrnice zavádí management sítě, který zahrnuje funkce:

3.6.1 Oživení komunikace (*Wake Up*)

Pokud je sběrnice v *Sleep módu*, tak kterákoliv jednotka si může vynutit probuzení, generováním budicího signálu. Budicí signál musí generovat dominantní úroveň v rozmezí 250µs až 5ms. Každý podřízený uzel musí umět detekovat čas o délce 150 µs a být připraven ke komunikaci během dalších 100ms. Vyšle-li podřízená jednotka žádost o probuzení, nadřízená jednotka musí být připravena s vysláním hlavičky (příkazového rámce) za účelem zjištění důvodu probuzení. Neděje-li se tak, podřízená jednotka pokus opakuje ve třech sekvencích a pak setrvá alespoň 1.5s. Situaci znázorňuje obrázek 40.



Obrázek 40 [14]: a) Budicí signál

b) Detekce a připravenost podřízené jednotky

c) Oživení komunikace podřízenou jednotkou

3.6.2 Uspání (Sleep mód)

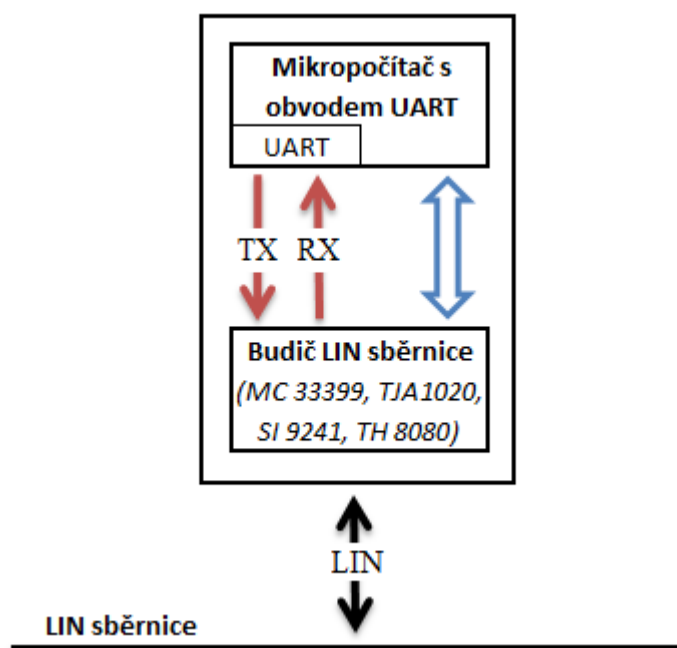
Uspání sítě proběhne na základě vyslání příkazu nadřazenou jednotkou nebo při nečinnosti sítě po dobu čtyř sekund. Nadřazená jednotka generuje prostřednictvím řídicího rámce s ID 0x3C s prvním datovým bajtem 0x00, který zašle všem jednotkám. Na obrázku 41 je vidět příkazový signál *Sleep módu*, přičemž bajty 2 až 8 podřízená jednotka ignoruje.

DATA 1	DATA 2	DATA 3	DATA 4	DATA 5	DATA 6	DATA 7	DATA 8
0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Obrázek 41: Příkaz Sleep módu

3.7 Hardwarové řešení jednotky LIN sběrnice

Podoba hardwarového terminálu ve funkci řídicí jednotky na LIN sběrnici je blokově znázorněna na obrázku 42.



Obrázek 42: Typické zapojení jednotky na LIN sběrnici (↔ řídicí signály)

- ✓ Mikroprocesor – Obsahuje události, vysílá a přijímá data na sběrnici LIN. V zásadě může být použitý libovolný mikroprocesor s integrovanou UART periferií a časovačem. Jsou vyvinuté i speciální mikroprocesory, které v sobě mají již implementovaný LIN budič např. PIC16C432 nebo PIC16C433 od výrobce Microchip. Dále existují procesory s integrovaným stabilizátorem, jejichž výhodou je možnost odpojení podřízených jednotek od napájení a dosažení úspory energie.
- ✓ Obvod UART – Hardwarový řadič sériové komunikace. Většinou je implementován v mikroprocesoru nebo je řešen softwarovou emulací UARTu.
- ✓ Budič (*transceiver*) LIN – Představuje fyzickou vrstvu protokolu, tj. převod TTL napěťové úrovně na úroveň 0 až 12 V. Z důvodu jednovodičového provozu budič potřebuje řízení emitovaného množství elektromagnetického rušení. Toho je převážně docíleno řízením rychlosti přeběhu signálu (*slew – rate*). Budič by měl obsahovat sleep mód, přičemž probuzení nastane od mikroprocesoru oživovacím signálem (*Wake Up*).

Poznámka: Konkrétní přehled integrovaných obvodů pro realizaci LIN jednotky je v příloze 4.

4 Sběrnice FlexRay

V dnešních automobilech se pro zpracování dat mezi řídicí jednotkou a akčními členy, senzory, či jinou jednotkou, používají nejčastěji komunikační sběrnice CAN a LIN. Ovšem stávající technologie se zdají být nedostatečné a tedy je založeno konsorcium FlexRay, které pracuje na standardu technologie X-by-Wire. Technologie X-by-Wire by měla vytlačit některé hydraulické systémy z automobilu a nahradit je vysoce inteligentními elektronickými systémy. Hovoří se hlavně o řízení (*Steer-by-wire*) a brzdění (*break-by-wire*).

4.1 Historie sběrnice FlexRay

2000 Založeno konsorcium FlexRay společnostmi BMV, DaimlerChrysler, Motorola a Philips. Později se připojily firmy Bosch, General Motors, VW a Freescale.

Poznámka: BMW je prvním výrobcem automobilů, který sběrnici FlexRay používá ve svých sériových vozech (*Adaptive Drive*). Lze ji nalézt v modelech BMW X5, BMW X6, BMW řady 7, BMW řady 5 GT a nyní nově také v BMW řady 5.

4.2 Vlastnosti sběrnice FlexRay

FlexRay je plně redundantní systém s velmi přesným časováním, řízením a ochranou proti všem možným chybám, které u tohoto systému mohou nastat. Tedy tato sběrnice je moderní vysokorychlostní vedení určené pro velmi náročné aplikace v automobilovém průmyslu z hlediska bezpečnosti. Je založený na přístupové metodě TDMA (*Time Division Multiple Access*), čili rozděluje komunikaci do časových slotů. Sběrnici dále charakterizují následující body:

- ✓ Přenosová rychlost 10Mb/s

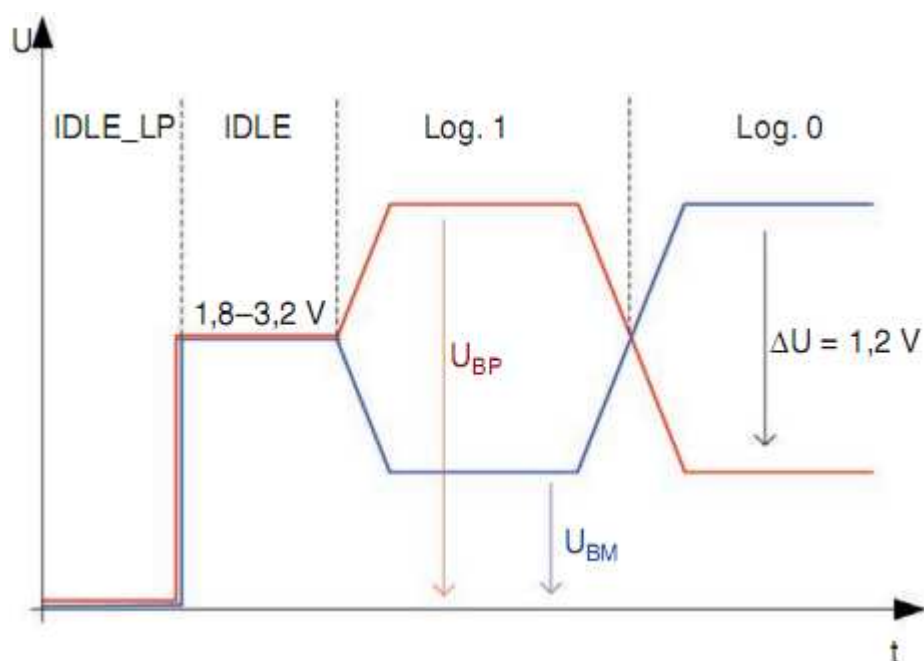
- ✓ Adresace až 2047 uzlů v jednom clusteru
- ✓ Jedná se o sběrnici se sériovým přenosem po dvou vodičích
- ✓ Vzdálenost mezi jednotkami na síti může být až 24m
- ✓ Dva shodné a nezávislé kanály pro každou jednotku
- ✓ Specifikace popisuje mnoho možností topologie
- ✓ diferenční zabezpečený přenos
- ✓ Detekce chyb na úrovni budičů

4.3 Fyzická vrstva

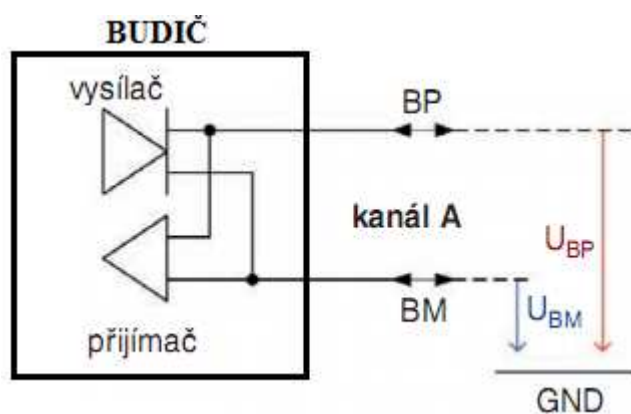
Fyzický vrstva FlexRay se povahově velmi blíží CAN sběrnici. Je realizována krouceným párem vodičů – BP (*Bus Plus*) a BM (*Bus Minus*) a nese diferenciální signál.

FlexRay pro svůj velmi bezpečný charakter je vybavena dvěma oddělenými redundantními kanály A, B, tzn. v případě selhání komunikace jednoho kanálu, nastupuje druhý. Podle obrázku 43 standard definuje následující stavy na síti:

- ✓ BUS_IDLE – Stav zapnuté sběrnice a však bez dat
- ✓ DATA_0 – Stav logické 1
- ✓ DATA_1 – Stav logické 0
- ✓ IDLE_LP – Úsporný režim (stav vypnuté sběrnice)



Obrázek 43: Elektrický signál na sběrnici FlexRay[22]



Obrázek 44: Vyznačené signály kanálem A [22]

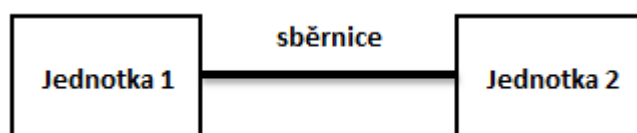
4.3.1 Zamezení rušňí na sběrnici FlexRay (EMC)

Podobně, jak u CAN sběrnice, viz kapitola 2.5.2

4.3.2 Topologie sítě

Sběrnice FlexRay disponuje širokou škálou propojení uzlů. Přičemž standard nakazuje dodržení maximálního zpoždění $2,5\ \mu\text{s}$ mezi dvěma jednotkami a maximální vzdálenosti mezi uzly 24 m. Uvádím alespoň základní topologie:

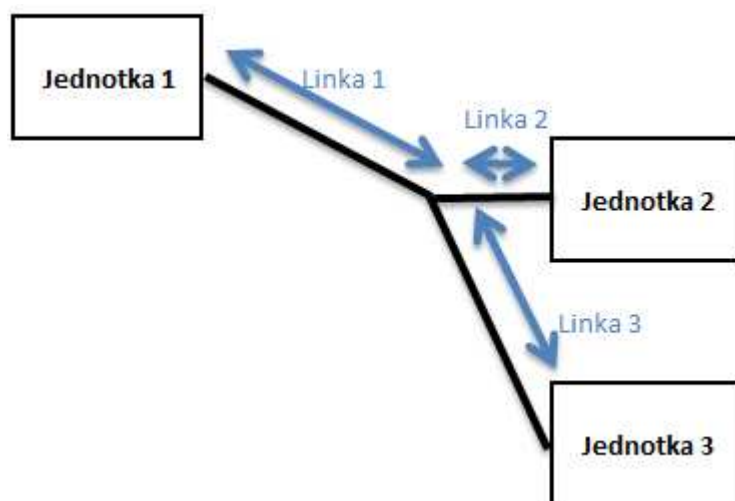
- ✓ Zapojení bod – bod (*Point – to – point*)



Obrázek 45: Zapojení bod – bod

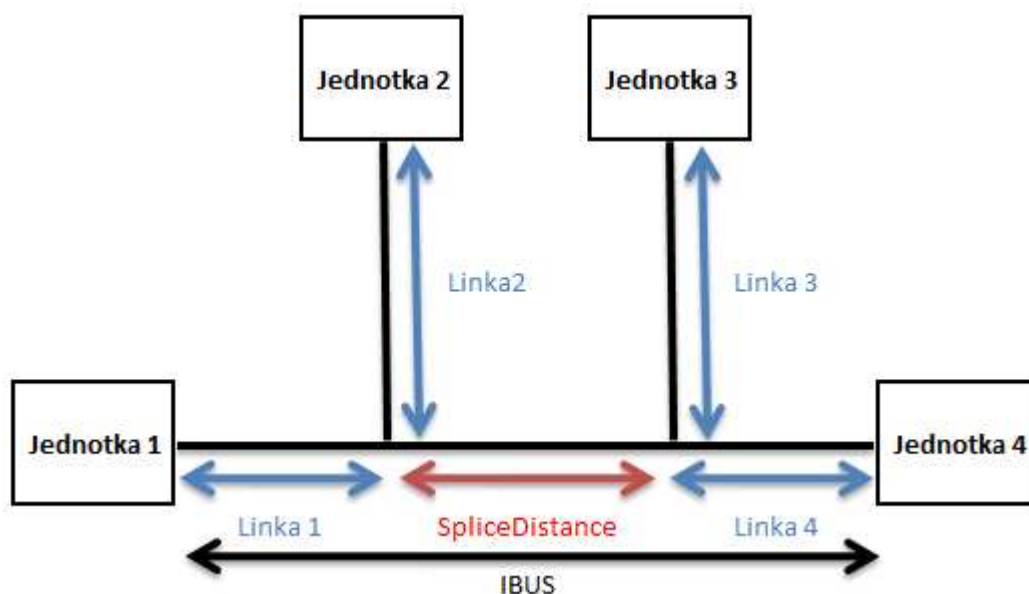
- ✓ Zapojení do hvězdy (pasivní síť)

Dovolená délka linky (*Stub*) je 12m



Obrázek 46: Zapojení do hvězdy (pasivní síť)

- ✓ Lineární zapojení (pasivní síť)
SpliceDistance minimálně 2,



Obrázek 47: Lineární zapojení (pasivní síť)

Specifikace definuje zapojení i s opakovací (např. *Active Star*) a různé propojení kanálu. Také povoluje kombinace uvedených základních topologií a pak takto vznikají tzv. hybridní topologie. Detailní informace popisuje literatura [20] a pak také [22].

4.4 Linková vrstva

4.4.1 Komunikační cyklus standardu

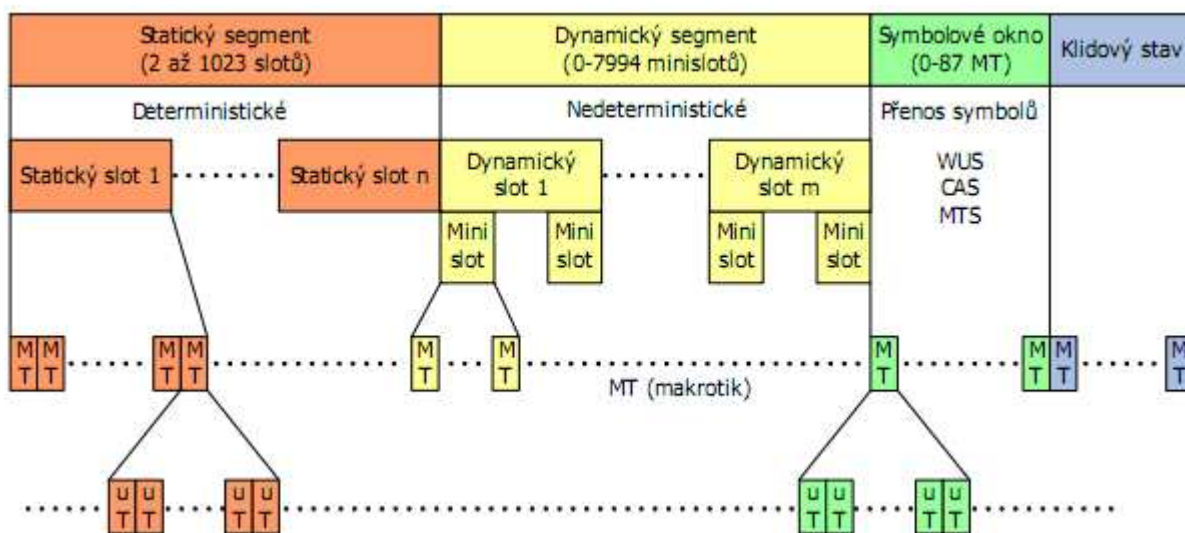
Komunikace v standardu FlexRay je založena na periodicky opakujících se komunikačních cyklech. Každý cyklus je rozdělený na časové sloty (TDMA). Jednotlivé sloty jsou přiděleny jednotlivým jednotkám.

Standard FlexRay používá dva druhy přístupových metod:

- ✓ TDMA (*Time Division Multiple Access*) – Je použitý pro vyslání prioritních zpráv se zaručenou latencí a je v komunikačním cyklu nazván, jako statický segment.

- ✓ FTDMA (*Flexible Time Division Multiple Access*) – Je použitý pro vyslání zpráv s nižší prioritou a v komunikačním cyklu je nazván, jako dynamický segment. Zde není zaručeno odeslání zprávy v každém komunikačním cyklu.

Poznámka: Detailní popis uvedených přístupových metod popisuje literatura [23].



Obrázek 48: Znázorněný komunikační cyklus [22]

Komunikační cyklus je znázorněn na obrázku 48, přičemž jednotlivé komunikační cykly se neustále opakují.

Segmenty se skládají z celistvého počtu časových slotů, přičemž v každém slotu je možno přenést jen jeden komunikační rámec. Počáteční konfigurace přiřazuje, které sloty v rámci jsou pro danou stanici.

Další popis je zaměřen na časové jednotky MT (macro tick) a μT (micro tick). Znázorněné intervaly (obrázek 48) statický slot (*static slot*) a mini slot (*minislot*) jsou násobkem celistvého počtu časových elementů macro tick (základní časové kvantum na sběrnici). Macro tick představuje nejmenší časový element globálního času na sběrnici a je na něm vystavěna synchronizace sběrnice. Musí být tedy stejná pro všechny jednotky v síti. Macro tick je násobkem celistvého počtu časových elementů μT . Počet μT v jedné MT, je různý pro všechny stanice, resp. je závislá na frekvenční časové základně, tak aby v průměru doba jednoho MT z pohledu pomyslného přesného času byla rovná pro všechny jednotky.

Statický segment musí obsahovat každý komunikační cyklus. Jak je vidět z obrázku 48 skládá se ze statických slotů, které jsou stejné dlouhé a vždy jich musí být celistvý počet. Jeden statický slot přináleží vždy jedné jednotce, která je oprávněná tímto slotem vyslat data (v situaci, když jednotka zpracovává více úloh najednou, může ji být přiděleno více statických slotů a takto je schopna odeslat více různých dat najednou). U příjmu dat je tomu jinak, zde může každá jednotka z libovolného slotu číst data. Komunikace statickým segmentem je úplně deterministická a vyhovující pro v úvodu zmíněné aplikace x-by-wire.

Dynamický segment nemusí být obsažen v komunikačním cyklu. Skládá se z celistvého počtu mini slotů, které vytvářejí dynamické sloty. Dynamické sloty mohou mít různou délku v závislosti na délce obsažených dat. Z toho plyne, že se jedná o nedeterministický přístupový princip. Což v důsledku vysílání velmi objemných dat (v počátečních slotech) se nemusí některým stanicím podařit odeslat data ani v aktuálně běžícím komunikačním cyklu a musí setrvat do doby dalšího komunikačního cyklu. Efektivita dynamického segmentu je v možnosti dynamického prodlužování slotů dle délky dat a tak dochází k šetření komunikačních prostředků.

Další části komunikačního cyklu je tzv. symbolové okno, jedná se o nepovinnou část a umožňuje vyslání tzv. symbolu (libovolná posloupnost bitů, třeba i nerespektující formát FlexRay datového bitu). Může nést diagnostické nebo řídicí informace, které jsou generované řadičem na popud aplikačního procesoru. Může být detekován i budičem a tedy představuje komunikační cestu uvnitř uzlu. S touto částí komunikačního cyklu se spíše počítá do budoucna s dalším rozšířením standardu FlexRay.

Poslední částí je tzv. klidový stav. Jedná se o povinnou část, která odděluje komunikační cykly. V této době, neprobíhá komunikace a sběrnice je v IDLE (klidový stav fyzického média).

4.4.2 Mechanismus Synchronizace jednotek

Všechny jednotky na FlexRay síti mají své zdroje hodinového signálu. Z důvodů nepřesnosti krystalu, které jsou způsobeny nejčastěji změnou teploty nebo stárnutím atd., nejsou frekvence časové základny stejné a stabilní. A vlivem akumulace těchto odchylek by došlo k porušení hranic všech komunikačních slotů a nastala by kolize (např. rámec by zasáhl

do slotů, který je přidělen jiné jednotce. Z tohoto důvodu je FlexRay standard vybaven synchronizačním mechanismem, který pracuje na principu tzv. synchronizačních rámců.

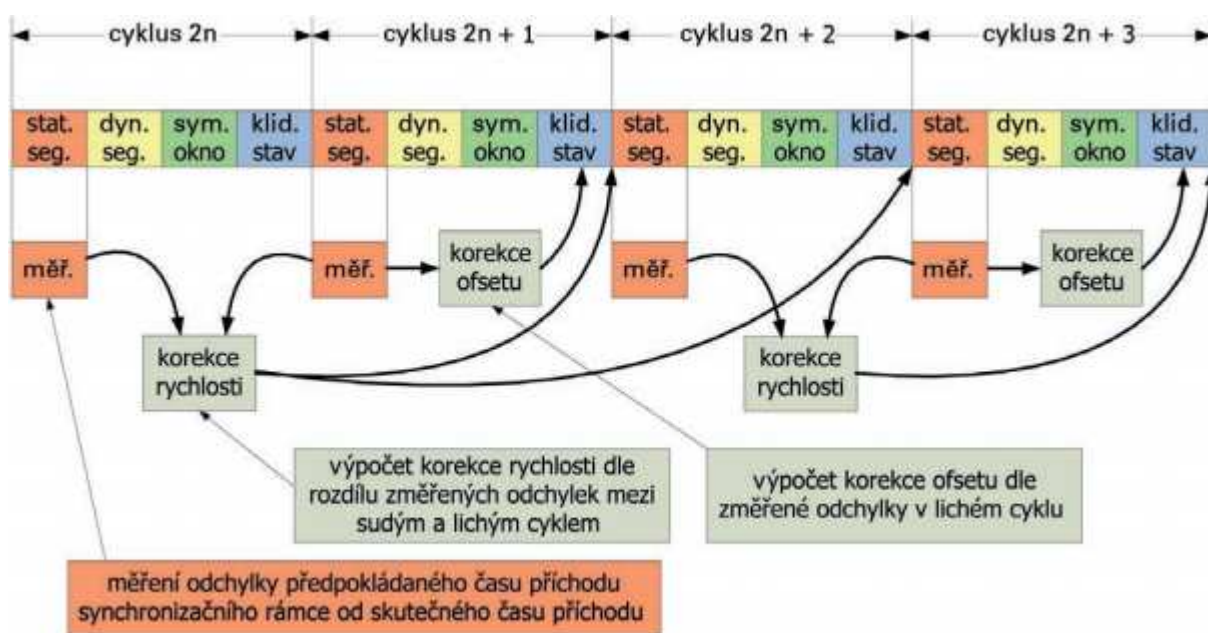
V každé síti se nachází jednotky, které synchronizují běh sběrnice. Takové to jednotky vysílají v statických slotech synchronizační rámce. Tyto rámce jsou přijímány všemi jednotkami na síti a pro každý přijatý synchronizační rámec počítá rozdíl (v mikritik) mezi dobou, kdy je očekáván jeho příchod okamžik, kdy je skutečně doručen. Měření odchylky je realizováno na základě použití algoritmu FTM (*Fault – Tolerant Midpoint*).

Podle počtu došlých synchronizačních rámců v statickém segmentu komunikačního cyklu se určí parametr „k“ (viz. tabulka 6)

Počet hodnot	Hodnota parametru k
1 až 2	0
3 až 7	1
>7	2

Tabulka 6: Určení parametru „k“

Odchytky jsou seřazeny podle velikosti, z těchto seřazených hodnot se odstraní „k“ max a „k“ min hodnota. Následně se ze zbylých hodnot určí z maxima a minima průměr. Tento průměr je brán, jako odchylka od globální časové základny a je provedena případná oprava s ohledem na korekci rychlosti a offsetu (offset je důsledek odlišných rychlostí časových základen). Princip synchronizačního mechanismu je zachycen na obrázku 49.



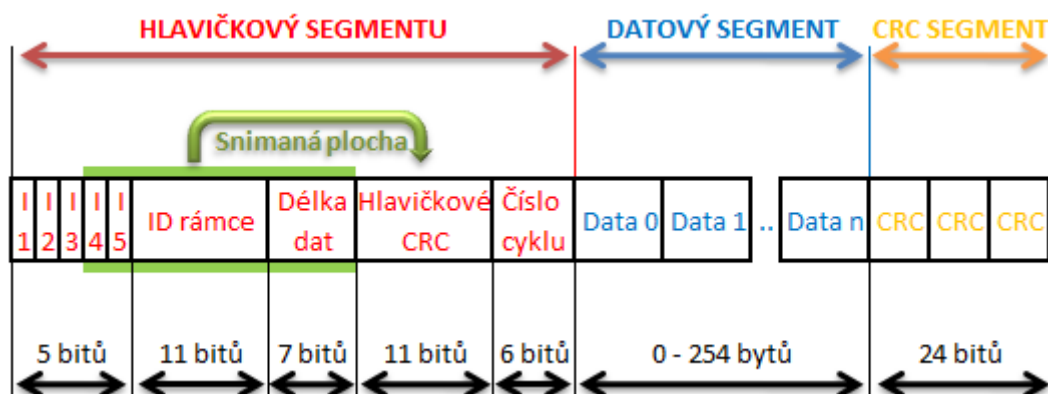
Obrázek 49: Znáznorněný synchronizační mechanismus 4 komunikačních cyklů [22]

4.4.3 Mechanismus startu sítě

Před započítím komunikačního cyklu (zapojení napájení a konfigurace jednotek) je třeba provést start sítě, pro tento účel jsou vyhrazeny dvě jednotky se statusem startu. Tyto jednotky vytvoří mezi sebou spojení a vyšlou startovací rámce, které jsou vyslány v patřičných slotech TDMA schématu.

4.4.4 Formát komunikačního rámce

Průběh komunikace a přenos dat mezi jednotkami je zprostředkován pomocí datového komunikačního rámce, který v sobě obsahuje přenášená data, řídicí prostředky, informace pro směrování, zabezpečení dat ad. Tento rámec se skládá z tří oddílů - hlavičkového segmentu (*Header Segment*), datového segmentu (*Payload Segment*) a CRC segmentu (*Trailer Segment*), viz obrázek 50. Datový rámec obsahuje NRZ kódování a před datovým rámcem se nachází startovací sekvence.



Obrázek 50: Formát datového rámce

Hlavičkový segment – je informační částí rámce

I1 – Reservovaný bit, nastaven v logické 0 (pro budoucí rozšíření standardu)

I2 – Indikátor preamble datového segmentu, aktivní v logické 1.

I3 – Indikátor nulového rámce

I4 – Indikátor synchronizačního rámce

I5 – Indikátor startovního rámce

ID rámce – uchovává číslo slotu, ve kterém je rámec vyslán (od 1 do 2047)

Délka dat – Informuje o délce datového segmentu (0 – 254 bajtů)

Hlavička CRC – Zabezpečení hlavičky CRC kódem. Standard popisuje, že CRC kód musí procesor spočítat a vložit s dalšími nastavujícími daty do registru komunikačního řadiče (počítáno z neměnné části hlavičky).

Číslo cyklu – Vypovídá o čísle cyklu z pohledu vysílací jednotky

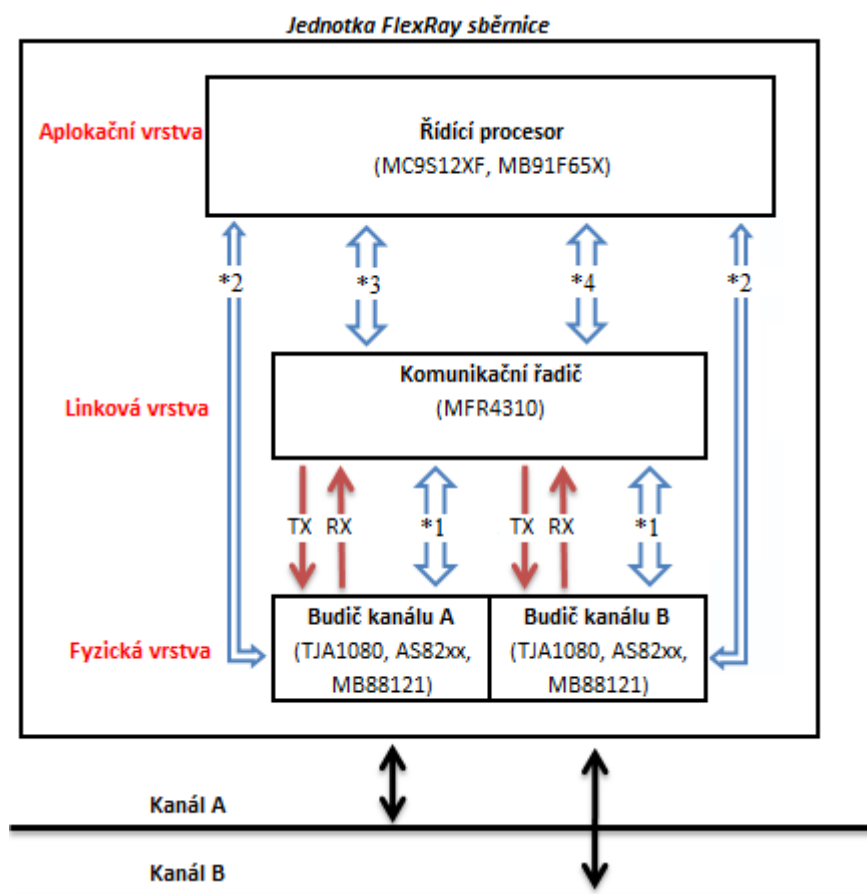
Datový segment – je částí, která přenáší požadovaná data

CRC segment – je vypočten řadičem, zabezpečuje data.

Detailnější informace k linkové vrstvě jsou v literatuře [21].

4.5 Hardwarové řešení jednotky - FlexRay sběrnice

Podoba hardwarového terminálu ve funkci řídicí jednotky na FlexRay sběrnici je blokově znázorněna na obrázku 51.



(*1 Řídicí signál, *2 Kontrolní data, *3 Konfigurační informace a stavové informace, *4 Komunikační data Rx a Tx,)

Obrázek 51: Typické zapojení jednotky na FlexRay sběrnici

Řídicí procesor (Host) – Zajišťuje implementace aplikační vrstvy, konfiguruje řadič při přivedení napájení a připravuje data, která mají být vyslány v průběhu komunikačního cyklu. Rozhraní mezi budičem a řídicím procesorem využívá dva diskrétní logické signály pro řízení provozního módu budiče a indikace chyb - STBN (Standby Not) a ERRN (Error Not).

Komunikační řadič (Communication Controller) – Realizuje funkce linkové vrstvy protokolu. Poskytuje řídicímu procesoru stavové informace a doručuje přijatá data

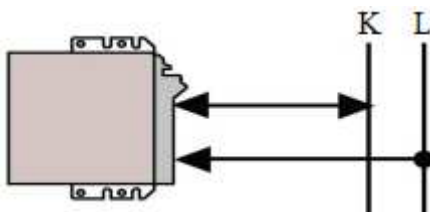
z komunikačního rámce. Na rozhraní mezi budičem sběrnice a řadičem se vyskytují tři elektrické signály, přičemž dva z nich jsou výstupy z komunikačního řadiče (TxD a TxEN) a jeden je vstup RxD. Komunikační řadič využívá signál TxD k přenosu aktuální signálové sekvence v momentě vyslání na komunikační kanál. Signál TxEN (*Transmit Data Enable Not*) pak indikuje řadič žádost budiči o vyslání dat na vysílací vodiče příslušného kanálu. Budič naopak využívá signál RxD k přenosu aktuálně přijaté datové sekvence do řadiče.

Budič sběrnice (*Bus Driver*) – Realizuje fyzickou vrstvu standardu FlexRay. V jednotce se nejčastěji vyskytují dva budiče sběrnice, jeden pro kanál A a druhý pro kanál B. Jejich volitelným prvkem, je tzv. hlídací člen (*Bus Guardian*). Úkolem hledacího členu je zabránit přístupu jednotky k sběrnici mimo časové sloty, které dané jednotce jsou určeny. Tato ochrana zabraňuje kolizím v případě poruchy jednotky.

Poznámka: Konkrétní průzkum integrovaných obvodů pro realizaci FlexRay jednotky je v příloze 5.

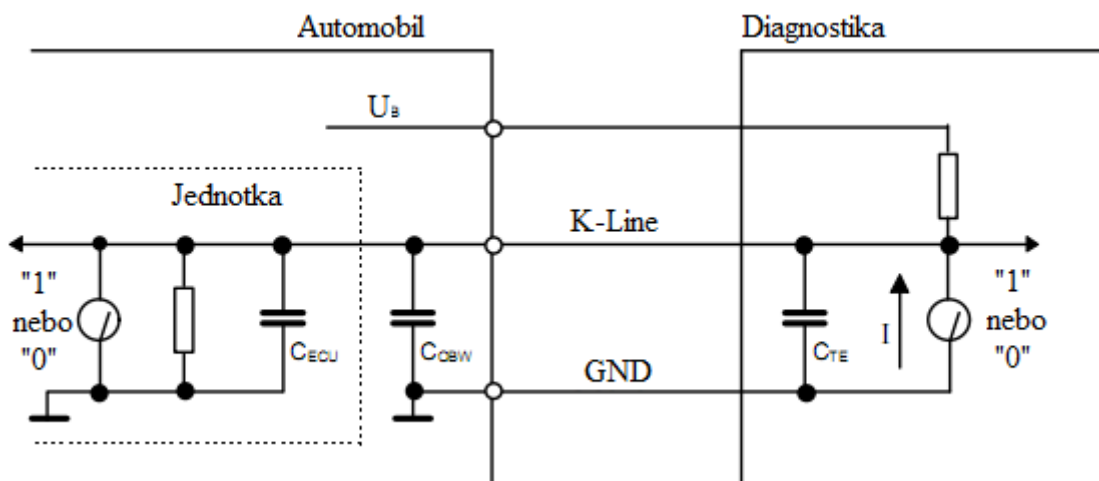
5 VEDENÍ K – Line, L - Line (ISO 9141)

Vedení K – Line a L- Line jsou diagnostické kanály, které definuje normou ISO 9141 – 2. V závislosti na roku vzniku automobilové řady se může setkat s kombinací K i L linky u starších automobilů, od roku 1996 se vyskytuje jenom K vedení a u nejnovějších modelů se lze setkat s KK diagnostickým vedením. Z obrázku 52 je patrné, že K linka komunikuje obousměrně a slouží pro posílání adres a dat. L vedení slouží pro souběžné (s K – Line) odesílání adres při inicializaci.



Obrázek 52: Směr komunikace obou linek

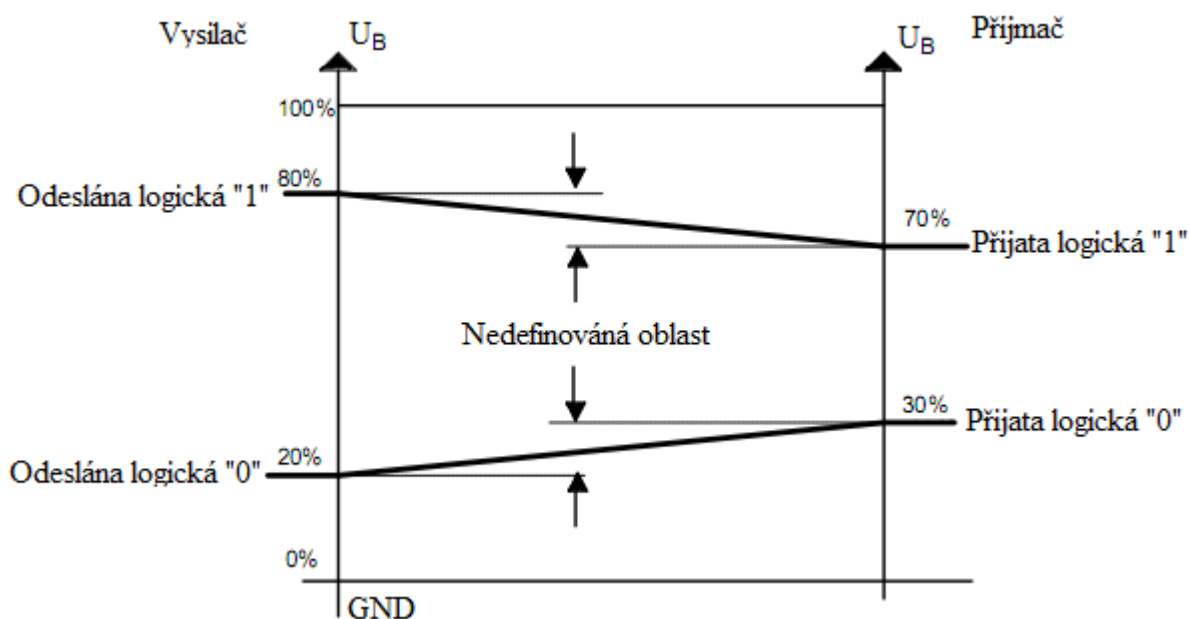
Zjednodušené zapojení jednotky a testovacího zařízení je vidět na obrázku 53.



Obrázek 53: Zjednodušené komunikační schéma podle ISO 9141

Výstupní napěťové úrovně se odvíjí od napětí na baterie U_B vůči kostře. A platí (obrázek 54):

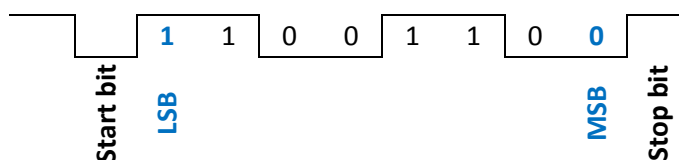
- ✓ Logické „0“ odpovídá menšímu napětí než 20% z napětí baterie pro vysílač a pro přijímač menšímu než 30% z napětí baterie.
- ✓ Logická „1“ odpovídá většímu napětí než 80% z napětí baterie pro vysílač a pro přijímač většímu než 70% z napětí baterie.



Obrázek 54: Mezní napěťové úrovně logických stavů

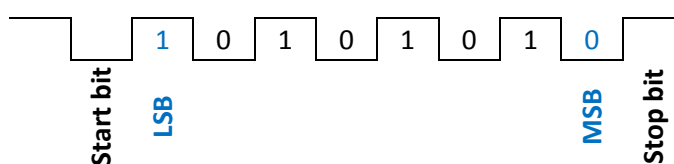
5.1 Inicializace komunikace

Podmínkou navázání spojení je udržení linky „K“ alespoň 2ms v jedničkovém stavu. Následně se vysílá osmi bitový rámec adresy s přenosovou rychlostí 5bit/s (jedná se pomalou inicializací) po „K“ a „L“ vedení. Hodnota adresového rámce je 33_H a je posílaná od nejvíc významného bitu MSB po nejméně významný bit LSB. Situace je znázorněná na obrázku 55.



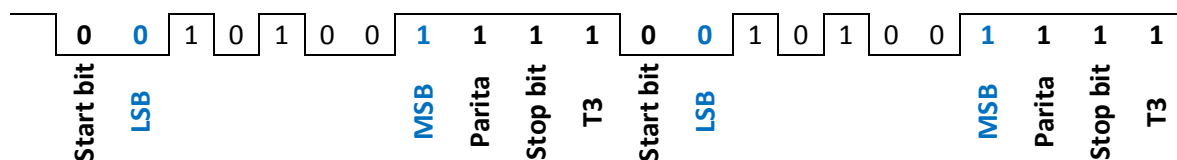
Obrázek 55: Adresový rámec

Po uplynutí času T1 (stav logická jedna), který nastane po adresovém rámci, je očekávány synchronizační bit z OBD II systému. Struktura rámce je naznačena na obrázku 56, přičemž se skládá ze střídajících se vysokých a nízkých stavů K linky při rychlosti 10,4kbit/s. Dodatečný čas na přizpůsobení rychlosti testovací jednotky je v protokolu zabezpečeno setrváním v jedničkovém logickém stavu po čas T2, za kterým OBD II systém odesílá následující bajty pro navázání spojení.



Obrázek 56: Synchronizační bajt

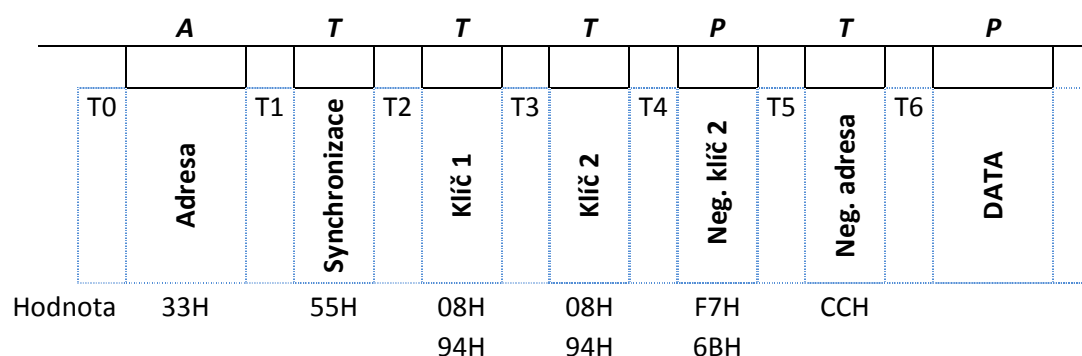
Následující bajty odesílá rovněž systém OBD II. Jedná se o dva 7 bitové rámce, označované literaturou [10], jako klíčová slova (*keyword*). Tyto klíčová slova slouží pro rozpoznání komunikačního protokolu a typu jednotky. Jsou znázorněny na obrázku 57. Ještě je třeba zmínit, že každé 7 bitové slovo je doplněno paritním bitem. Klíčová slova jsou oddělena jedničkovým bitem, který trvá čas T3.



Obrázek 57: Formát klíčových slov

Poznámka: Literatura [10] uvádí, že normalizační výbor FAKRA, sídlící v Německu, přiděluje výrobcům automobilu unikátní klíčová slova. Například čísla od 71 do 120 jsou přiděleny automobilce Opel.

Tester po přijetí druhého klíčového slova a uplynutí jedničkového stavu, který trvá čas T4, odesílá negované druhé klíčové slovo jednotce automobilu. Následně se musí udržet jedničkový stav po dobu T5. A pak je odeslána systémem OBD II negovaná hodnota adresy (CC_H). Struktura celé sekvence je znázorněna na obrázku 58.



*Směr komunikace: A – automobil, T – Tester

Obrázek 58: Inicializace přenosu v systému ISO 9141 – 2

5.2 Datový rámeček

Komunikace s ohledem na přenos diagnostických dat je spojena v pakety, které jsou nazývány informacemi. Skládají se z 5 až 11 bajtů, přehled je znázorněn na obrázku 59. K obrázku je třeba doplnit, že jednotlivé datové informace jsou oddělené časem P2 a P3 v závislosti na směru přenosu dat (do automobilu / testeru). A jednotlivé bajty v datové informaci, jsou odděleny dobou P1 (bajty z automobilu) a P4 (bajty z testeru). Na obrázku 59 jsou uvedeny datové rámce ISO 9141 a ISO 14230. Na obrázku je dále vidět SAE J1850 datový rámeček, který používány v amerických značkách automobilu (Ford).

Hlavička			Datové pole							Kontrolní bajty
Priorita	Cílová adresa	Zdrojová adresa	1	2	3	4	5	6	7	ERR
<i>Diagnostický požadavek při přenosové rychlosti 10,4kbit/s podle ISO 9141 - 2, SAE J1850</i>										
68h	6Ah	FXh	Datové pole (maximálně 7 bajtů)							ERR
<i>Diagnostická odpověď při přenosové rychlosti 10,4kbit/s podle ISO 9141 - 2, SAE J1850</i>										
48h	6Bh	drojová adres	Datové pole (maximálně 7 bajtů)							ERR
<i>Diagnostický požadavek při přenosové rychlosti 10,4kbit/s podle ISO 14230 - 4</i>										
11DDDDDDb	6Ah	FXh	Datové pole (maximálně 7 bajtů)							ERR
<i>Diagnostická odpověď při přenosové rychlosti 10,4kbit/s podle ISO 14230 - 4</i>										
10DDDDDDb	FXh	drojová adres	Datové pole (maximálně 7 bajtů)							ERR
ERP - bajt kontrolní sumy CRC										
DDDDDD - délka dat (jen ISO 14230 - 4)										

Obrázek 59: Datové diagnostické formáty v systému OBD II / EOBD

6 Přehled ostatních vedení

6.1 Datová sběrnice MOST

MOST (Media Oriented System Transport) je systém vyvinutý pro multimediální funkce automobilu. Jedná se o optickou linku. V roce 2001 tato sběrnice byla poprvé použita u vozu BMW řady 7.

6.2 Datová sběrnice ACP

ACP (Audio Control Protocol) je určena pro audio aplikace a vestavěný telefon

6.3 SAE J1850

Diagnostická linka amerických automobilů, jako je například automobil Ford. Je standardem OBD II.

- PWM (*Pulse width Modulated*)

- 41.6 kbps

- maximální délka vedení 40 m a 32 jednotek

- diferenciální signál

- VPW (Variable Pulse Width)

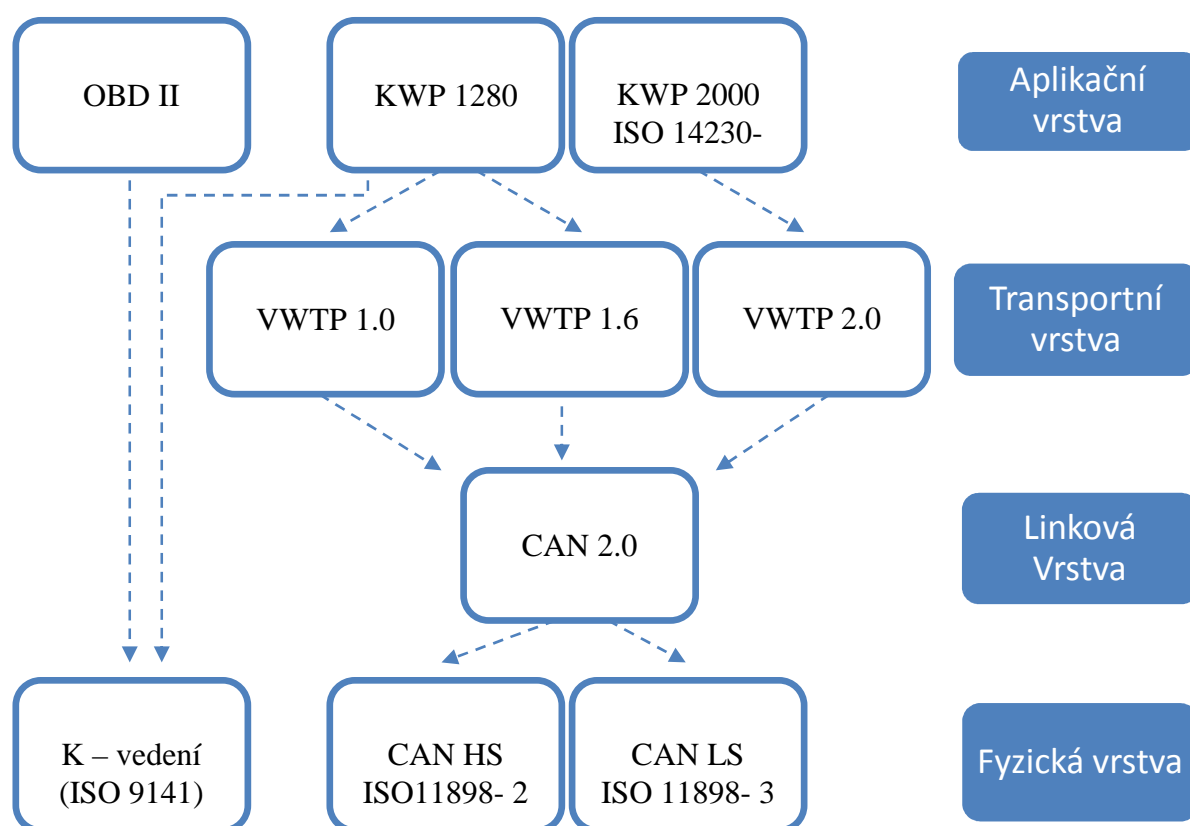
- 10.4 kbps

- maximální délka vedení 40m a 32 jednotek

- jednovodičové provedení

7 Diagnostické protokoly

V další části si rozebereme diagnostické protokoly pracující na transportní a aplikační vrstvě v komunikaci automobilu, přičemž se zaměříme na protokoly v evropských automobilech (obrázek 60).



Obrázek 60: Přehled komunikačních protokolů

7.1 Transportní vrstva

Jak je zřejmé z obrázku 60 transportní vrstva je použita při přenosu dat CAN sběrnici. Princip činnosti spočívá v rozkladu dat z vyšší aplikační vrstvy na CAN datové rámce

zpracovatelná linkovou vrstvou a dále naopak skládá CAN datové rámce tak, aby byly detekovatelná aplikační vrstvou. Mezi evropské zmapované protokoly transportní vrstvy řadíme VWTP 1.0, VWTP 1.0, VWTP 1.6 a VWTP 2.0 (koncern Volkswagen – Volkswagen, Seat, Audi, Škoda).

7.1.1 VWTP 1.6

Volkswagen transport protokol 1.6 je protokolem transportní vrstvy, který zajišťuje komunikaci CAN sběrnice s protokolem aplikační vrstvy KW 1281. Zabezpečuje pouze komunikaci mezi řídicími jednotkami v automobilu (most mezi K vedením a CAN sběrnici). Tento protokol definuje následující činnosti:

- ✓ Nedefinuje některé stavy a příkazy
- ✓ Vytvoření dynamického kanálu mezi dvěma CAN jednotkami (na základě adresy)
- ✓ Kontrola spojení potvrzením každého bloků paketů
- ✓ Oprava vzniklých chyb

Poznámka: Nepoužívá se pro diagnostiku (Převádí jen požadavky mezi K vedením, resp. KW1281 a protokolem CAN).

Princip činnosti

Na počátku transportu je třeba vytvořit dynamický kanál, pomocí odeslání instalačního kanálového (*Channel Setup*) paketu.

ID	Místo	Kód	Identifikátor kanálu
0 - 10 bit	1. Byte	2. Byte	3. Byte

Obrázek 61: Paket pro vytvoření dynamického kanálu – VWTP 1.6

ID – identifikace vysílající jednotky

Místo – adresa cílové jednotky

Kód – provedena operace

Identifikátor kanálu – identifikátor odpovídající jednotce po nastavení kanálu

Stejný tvar má i potvrzovací paket (*Channel Acknowledge*)

Vytvořeným kanálem mohou být odesílané následující pakety:

- ✓ DT – datový paket zabezpečující přenos z aplikační vrstvy tj. KW1281
- ✓ ACK – potvrzení přijetí datového paketu, interval po kterém jsou pakety potvrzeny je nastaveno v CS paketu
- ✓ CS (*Connection Setup*) – nastavení dynamických parametrů spojení
- ✓ CA (*Connection ACK*) – potvrzení CS paketu
- ✓ DC – ukončení spojení, resp. zrušení kanálu. Každá jednotka má určitý počet kanálu

Udržení komunikace zabezpečuje protokol vyšších vrstev – aplikační vrstva (KW 1281).

7.1.2 VWTP 2.0

Volkswagen transport protokol 2.0 je rovněž protokolem transportní vrstvy, který zajišťuje komunikaci CAN sběrnice (High-speed, 0.5Mbps) s protokolem KW 2000. Tento protokol definuje následující činnosti:

- ✓ Vytvoření dynamického kanálu mezi dvěma CAN jednotkami (na základě adresy)
- ✓ Kontrola spojení potvrzením každého bloků paketů
- ✓ Oprava vzniklých chyb

Princip činnosti

Na počátku transportu je třeba vytvořit dynamický kanál, pomocí odeslání instalačního kanálového (*Channel Setup*) paketu.

ID	Místo	Kód	Tx - ID Low	Info Tx	Tx - ID High	Rx-ID Db Low	Info Rx	Rx - ID High	Apl. typ
10-0 b	1. Byte	2. Byte	3. Byte	7-3 bit 4. Byte	2-0bit	5. Byte	7-3 bit 6. Byte	2-0bit	7. Byte

Obrázek 62: Paket pro vytvoření dynamického kanálu – VWTP 2.0

ID – identifikace vysílající jednotky,

Místo – adresa cílové jednotky

Kód – provedena operace

Tx-ID – požadavek = 0, Odpověď = identifikátor jednotky, která začala vysílat

Info-Tx – bit 3 = rezerva, bit 4 = požadavek (požadavek = 1, Odpověď = 0), bit 5,6,7 = 0

Rx-ID – požadavek (Identifikátor jednotky, která komunikaci zahájila), odpověď (Identifikátor jednotky, která reagovala na započetí komunikace)

Info-Rx – bit 3 = rezerva, bit 4 = 0, bit 5,6,7 = 0

Stejný tvar má i potvrzovací paket (*Channel Acknowledge*)

Vytvořeným kanálem mohou být odesílaný tyto pakety:

- ✓ DT – datový paket zabezpečující přenos z aplikační vrstvy tj. KW 2000
- ✓ ACK (*Acknowledge*) – potvrzení přijetí datového paketů, interval potvrzování, je nastaven v CS
- ✓ CS (*Connection Setup*) – nastavení dynamických parametrů spojení (vysílán jednotkou, která zahájila komunikaci)
- ✓ CA (*Connection ACK*) – potvrzení CS paketu
- ✓ CT (*Connection test*) - test spojení. Odpověď přijíací jednotky je CA paket
- ✓ BR (*Break*) – žádost o přerušeni komunikace
- ✓ DC (*Disconnect*) – ukončení spojení, resp. zrušení kanálu. Každá jednotka má určitý počet kanálu

Paket	7. bit	6. bit	5. bit	4. bit	3. bit	2. bit	1. bit	0. bit
DT	0	0	AR	EOM	SN			
ACK	1	0	RS	1	SN			
CS	1	0	1	0	0	0	0	0
CA	1	0	1	0	0	0	0	1
GS	1	0	1	0	0	0	1	0
CT	1	0	1	0	0	0	1	1
BR	1	0	1	0	0	1	0	0
DC	1	0	1	0	1	0	0	0

Obrázek 63: Formát jednotlivých typů kanálových paketů – VWTP 2.0

AR (*Acknowledge Request*) – pokud se tento bit rovná nule, odesílatelem je žádáno potvrzení posílaného paketů

EOM (*End of Message*) – konec zprávy (logická hodnota 0)

RS (*Receive status*) – ACK paket s RS nastavený v logické nule

SN (*Sequence number*) – Sekvenční číslo sloužící pro kontrolu, každým paketem se inkrementuje.

Po vytvoření dynamického kanálu tester odesílá paket CS (*Connection Setup*), při kterém se nastaví parametry kanálu (např. časování). Dále čeká na potvrzovací paket CA. Za předpokladu, že tester momentálně nekomunikuje, musí odesílat testovací paket (*Connection Test*). Takto je udržovány komunikační kanál, resp. komunikace. Odpověď dané jednotky je zde CA paket.

Datový rámec slouží pro výměnu požadavku mezi testovacím zařízením a jednotkou. Požadavek na potvrzení zprávy přijímací jednotkou může být odesílatelem generován v kterémkoliv rámci, přičemž před další komunikaci musí být odpovězeno pomocí paketu ACK.

Nestačí-li na požadavek 6 bajtů, je možné odeslat další datové rámce s inkrementovaným sekvenčním číslem SN.

Ukončení spojení resp. zrušení dynamického komunikačního kanálu je provedeno pomocí paketu DC (*Disconnect*). Kanál je nutné ukončit, protože každá jednotka má omezený počet možných kanálů.

7.2 Aplikační vrstva

Aplikační vrstva zpracovává význam dat, tzn., generuje příkazy a přijímá odpovědi. Mezi evropské popsané protokoly aplikační vrstvy patří KWP 1280 a KWP 2000, která jsou připojena buď CAN sběrnici přes transportní protokol, nebo přímo K vedením.

7.2.1 KW 1281

Protokol KW 1280 (*Keyword protocol 1281*) byl používán, až do roku 1998 všemi jednotkami koncernu VW. Předchůdce tohoto protokolu byl protokol Bosch KW71, přičemž uvedený protokol KW 1281 je zpětně kompatibilní s KW71.

Princip komunikace je založen na pravidelném střídání komunikace (tester – jednotka – tester – jednotka atd.). Komunikace probíhá v tzv. blocích. Uvedený protokol se používá přímo s linkovou vrstvou K - vedení (ISO1941) nebo v kombinaci s VW transportním protokolem 1.6 a linkovou vrstvou CAN protokolu 2.0 (ISO11898).

KW 1281 protokol stanovuje tyto základní příkazy:

- ✓ Identifikace jednotky
- ✓ Čtení RAM, ROM a EEPROM, zápis do RAM
- ✓ Test Akčních členů
- ✓ Mazání adaptačních hodnot
- ✓ Čtení měřených hodnot
- ✓ čtení chyb paměti
- ✓ Kódování jednotek atd.

7.2.2 KWP 2000 (ISO 14230)

Protokol KWP 2000 (*Keyword protocol 2000*) se začíná používat od roku 1998 a nahrazuje KW1281. Je významný následujícími odlišnostmi:

- ✓ Stála komunikační rychlost (10400 bps)
- ✓ ECU musí odpovídat klíčem 2000 – 2099

Princi činnosti spočívá ve výměně zpráv mezi jednotkou a diagnostikou (požadavek – odpověď), přičemž požadavky generuje jen testovací zařízení.

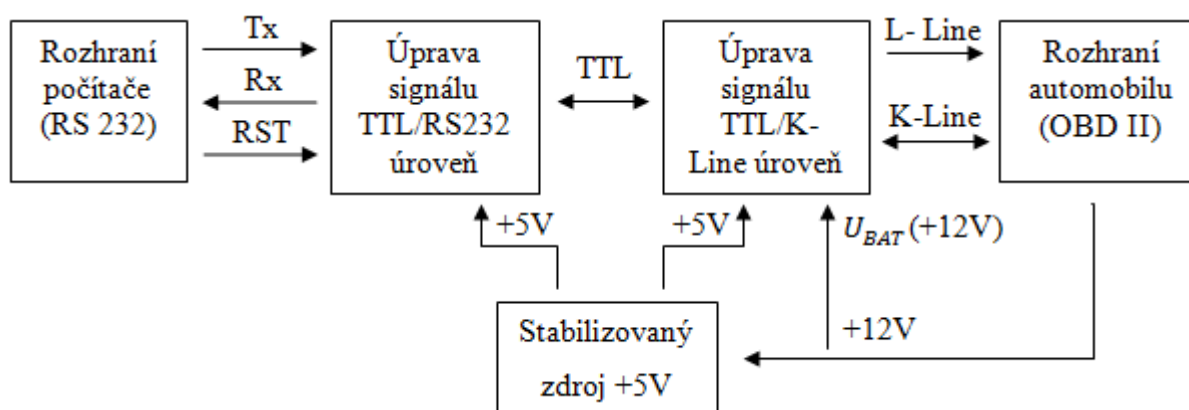
KW 2000 protokol stanovuje tyto základní obecné příkazy

- ✓ Identifikační služby jednotek
- ✓ Sjímaní naměřených hodnot
- ✓ Sjímaní paměti závad, mazání chyb
- ✓ Testování akčních členů
- ✓ Doplnkové funkce – Nastavení parametrů jednotky atd.
- ✓ Synchronizace komunikace a sjednocení parametrů
- ✓ Nahrávání (update) firmware jednotky

Další informace jsou k dispozici v literatuře [26] a [27].

8 Realizace hardwarového terminálu (K – line interface)

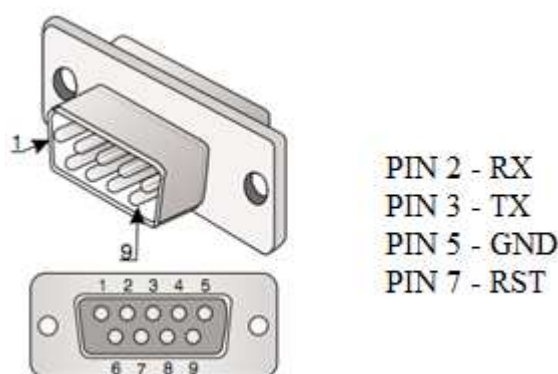
Výše jsme si teoreticky rozebrali diagnostické rozhraní automobilu po K vedení. V následující části si popíšeme realizaci K – Line a L - Line interfejsu (ISO 9141). Přičemž se budeme zabývat hardwarovým řešením, jež je blokově znázorněn na obrázku 64. Podporuje diagnostické software pro RS 232 (Vag - Com, ProDiag 1.0x, Lemmiwinks ECU Tuning a další).



Obrázek 64: Blokové schéma K – line interfejsu

8.1 Převodník RS 232 / TTL (TTL / RS 232)

Převedení úrovní RS 232 na TTL a na opak je realizován pomocí známého obvodu MAX232 od firmy Maxim. Tento obvod vyžaduje externí kondenzátory, které plní funkci nábojové pumpy, pro dosažení napěťové úrovně RS232. Zapojení bylo provedeno dle doporučení výrobce. Na obrázku 65 jsou znázorněné potřebné piny.



Obrázek 65: Konektor DB9

8.2 Převodník K, L line / TTL (TTL / K, L line)

Tento blok se dá řešit několika způsoby:

- ✓ Úprava signálu pomocí tranzistorů (rušení na větší vzdálenosti, chybí zabezpečení např. proti zkratu na kostře atd).
- ✓ Úprava signálu pomocí tranzistoru a optočlenů (rovněž chybí zabezpečení)
- ✓ Úprava signálu pomocí originálních IO postavených na ISO 9141 (velmi výhodné z hlediska ceny a zpracovanosti ochrany)

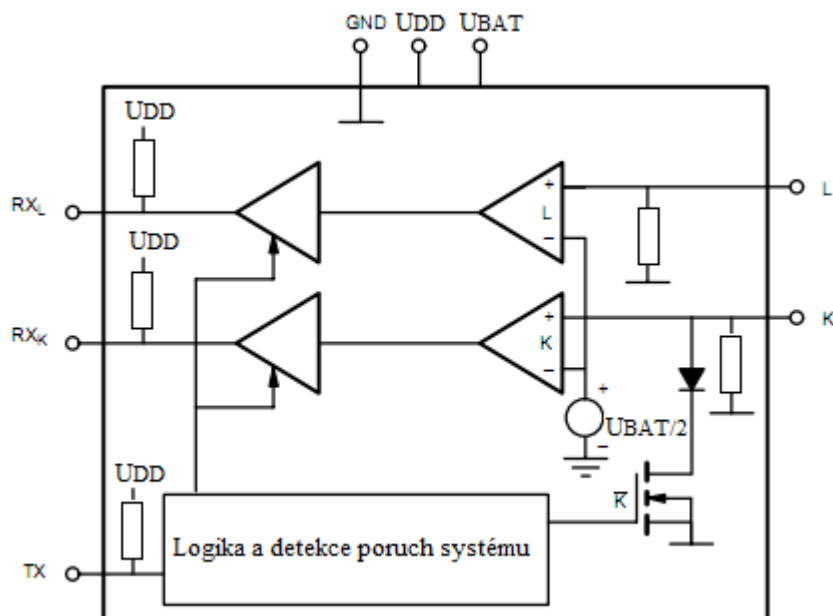
Byl proveden průzkum trhu a byly objeveny následující možné obvody: L9637D od firmy STMicroelectronics, MC33290 od firmy Motorola/Freescale a obvody SI9243AEY, SI9241AEY od společnosti Vishay

Pro realizaci jsem si zvolil obvod firmy Vishay SI9243AEY. Jeho výhodou je výrobcem definovaná přenosová rychlost a má v sobě integrovanou i linku L. Je dostupný v GES elektronik a cenově výhodný.

8.2.1 Popis obvodu SI9243AEY

Obvod SI9243AEY je sběrniceový přijímač a vysílač navržený pro obousměrnou komunikaci v automobilových diagnostických aplikacích. Jeho princip činnosti spočívá v porovnávání vstupů K a L s napětím baterie uvnitř obvodu. A pokud na K nebo L vstupu je

napětí menší než 50% U_{BAT} pak RxK či RxL výstup bude v logické 0. A naopak, pokud na K nebo L vstupu bude napětí větší než 50% U_{BAT} pak RxK či RxL výstup bude v logické 1.

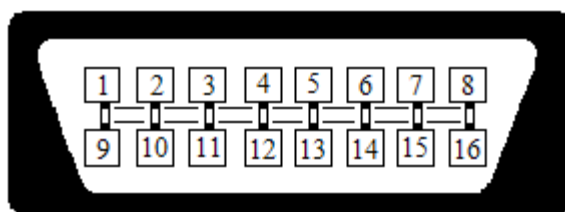


Obrázek 66: Zjednodušená vnitřní struktura IO SI9243AEY [28]

Obvod disponuje odesílacím a přijímacím módem. Pokud na TX pin je přivedena logická 0 aktivuje se mód odeslání dat (komunikace od testeru), je to způsobeno invertováním přivedené logické 0 a tak sepnutím MOSFET tranzistoru, který nastaví pin K do nízké úrovně. Naopak pokud je Tx pin je vysoké úrovně, tak vnitřní tranzistor je otevřený a může nastat přijímací mód (komunikace od automobilu).

Obvod je schopný detekovat zkrat linky na kostru a teplotní překročení. Procesor během odesílacího režimu vyhodnocuje RxK a Tx signál, když oba signály se zrcadlí, tak není vyhodnocena chyba. V případě detekce poruchy K pin zůstane ve vysoké impedanci a RxK následně taky.

8.2.2 Zásuvka EOB



Zásuvka EOBD

Obsazení pólů diagnostického konektoru - zásuvka EOBD

1, 3, 8, 9, 11, 12, 13 – neobsazeno	7 – K-Line
2 – komunikační protokol SCP +	10 – komunikační protokol SCP -
4 – kostra vozidla	14 – CAN <u>Low</u>
5 – kostra signálu	15 – L-Line
6 – CAN <u>High</u>	16 – +12V

8.3 Ověření funkčnosti Hardware

Realizovaný Hardwarový terminál sériového komunikačního vedení, byl ověřený nejprve na základě proměření požadovaných napěťových úrovní na jednotlivých blocích. Po té byla funkce otestována na automobilu Fabia, Octavia II a laboratorním panelu řídicích jednotek. Následně byla sestavena laboratorní úloha a byl vyhotoven protokol (vzor).

9 Závěr

Cílem této diplomové práce bylo vytvoření detailní rešerše z oblasti automobilových komunikačních vedení a zpracování věcných opěrných bodů, pro případ dalšího užšího zkoumání této problematiky. V oblastech, kde nedovolil rozsah práce nebo odbornost zkoumané části, byl text doplněn odkazem na příslušnou literaturu.

V rámci této diplomové práce byl realizován hardwarový terminál pro zpracování dat z automobilové komunikační sběrnice. Navrhnutý hardware bude používán v praktické části výuky předmětu automobilové elektroniky, pro nějž byla také vypracována laboratorní úloha.

Realizovaný hardware je vybaven diagnostickou zástrčkou standardu EOBD pro připojení k automobilu. Na straně počítače je použito RS232 rozhraní. Případná absence komunikačního rozhraní RS232 v PC, je řešena jiným studentem v rámci bakalářské práce, kde student řeší FTDI převodník.

Další vývoj diplomové práce v budoucnu je možný z hlediska rozšíření jednotlivých teoretických záležitostí a věnování se řešení a návrhu konkrétní řídicí jednotky. Například realizace FlexRay nebo CAN Bus jednotky, kde bude nutné se zabývat programováním mikroprocesorů.

V tomto akademickém roce je například realizovaná diplomová práce zabývající se světelným systémem automobilu, dále ve vzdálenější době by mohl být realizován adaptivní tempomat s FlexRay Bus. V uvedených záležitostech, lze využít, resp. lze navázat na tuto diplomovou práci při řešení komunikace s ostatními jednotkami.

Seznám požitých literatury

- [1] Polák, K.: *Sběrnice CAN* [online]. Brno, 2003. Ústav telekomunikací, VUT FEKT v Brno. Dokument je dostupný na URL <http://www.elektrorevue.cz/clanky/03021/index.html>
- [2] Jakubicki, D., Kalecki, M.: *Controller Area Network – Informacje* [online]. Wrocław, 2007. Wydział Elektroniki Wrocławskiej. Dokument je dostupný na URL <http://free.of.pl/c/can1/>
- [3] Jakl, P., Čermák, O.: *Elektronické systémy pro diagnostiku vozidel v dlouhodobých zkouškách*. 2007. Prezentace Simply Clever Škoda Auto
- [4] Novák, J., Kocourek, P.: *Fieldbus* [online]. Praha, 1998. ČVUT v Praze. Dokument je dostupný na URL <http://fieldbus.feld.cvut.cz/>
- [5] Vlk, F.: *Automobilová elektronika 2 – Systémy řízení podvozků a komfortní systémy*, 1. vydání. Brno, 2006. ISBN 80-239-7062-3
- [6] Drábek, J.: *Sběrnice používané v oblasti programovatelných automatů*. Brno, 2008. Bakalářská práce na VUT v Brně. Vedoucí bakalářské práce Tomáš Marada
- [7] Bryx, R.: *Monitorování sběrnice CAN osobního automobilu Fabie*. Ostrava, 2005. Diplomová práce na VŠB TU - Ostrava. Vedoucí diplomové práce Petr Kočí.
- [8] Bettelheim, R.: *CAN Controller Area Network*. 2005. Freescale Technology presentation.
- [9] Chlebiš, P.: *Konstrukce elektronických zařízení*. Ostrava, 2009. Prezentace na VŠB TU Ostrava.
- [10] Grzeszczyk, R., Kowalewski, D.: *Odbiór i interpretacja informacji z pokładowych systemów diagnostycznych OBD* [online]. Gdańsk, 2000. Dokument je dostupný na URL www.ilot.edu.pl/KONES/2000/01/16_JOK2000.pdf

-
- [11] Myszkowski, S., Mazurek, S.: *Pokładowe systemy diagnostyczne pojazdów samochodowych OBD*, 3. wydanie. Str 338 - 344. ISBN 978-83-206-1633-0
- [12] Sutorý, T: *LIN (Local Interconnect Network)* [online]. Brno, 2004. Ústav radioelektroniky, VUT FEKT v Brně. Dokument je dostupný na URL <http://www.elektrorevue.cz/clanky/04012/index.html>. ISSN 1213-1539.
- [13] Vojáček, A.: *LIN (Local Interconnect Network)* [online]. 2004. Dokument je dostupný na URL <http://www.automatizace.hw.cz/mereni-a-regulace/ART133-lin--local-interconnect-network.html>
- [14] Machálka, L.: *Hardwarové implementace komunikačního protokolu LIN (Local Interconnect Network) osmibitovými mikroprocesory a jejich srovnán.* Brno, 2008. Ústav telekomunikace, VUT FEKT v Brně. Elektrorevue, ISSN 1213-1539.
- [15] LIN Consortium. *LIN Specifikation Package Revision 1.3* [online]. Německo, 2002. Dokument je dostupný na URL: <http://www.lin-subbus.org/>
- [16] LIN Consortium. *LIN Specifikation Package Revision 2.0* [online]. Německo 2003. Dokument je dostupný na URL: <http://www.lin-subbus.org/>
- [17] LIN Consortium. *LIN Specifikation Package Revision 2.1* [online]. Německo 2006. Dokument je dostupný na URL: <http://www.lin-subbus.org/>
- [18] Hank, P., Johnk, E.: *Application Note - SJA1000 Stand_alone CAN controller (AN97076)* [online]. Hamburg, 1998. Systems Laboratory Hamburg, Dokument je dostupný na URL: http://www.nxp.com/documents/application_note/AN97076.pdf
- [19] *Magistrala CAN*, Polska [online]. 2000. Czasopismo Elektronika praktyczna 1/2000, Dokument je dostupný na URL: <http://www.elektronik.zst.edu.pl/files/can4.pdf>
- [20] FlexRay Consortium: *Electrical Physical Layer Specification, Version 2.1B* [online]. 2006. Dokument je dostupný na URL: www.FlexRay.com

-
- [21] FlexRay Consortium: *FlexRay Protocol Specification, Version 2.1 A* [online]. 2005. Dokument je dostupný na URL: www.FlexRay.com
- [22] Malinský, J., Kocourek, P., Novák, J.: *Komunikační standard FlexRay*. Praha, 2009. ČVUT v Praze. Časopis Sdělovací technika 9/2009, ISSN 0036 - 9942
- [23] Waraus, D.: *Měřicí systém na bázi sběrnice FlexRay*. Praha, 2008. Diplomová práce na ČVUT v Praze. Vedoucí diplomové práce Jan Malinský.
- [24] Zimmermann, W., Schmidghall, R.: *Magistrala danych w pojazdach*. 2009, str. 72 – 89. ISBN: 978-83-206-1698-9.
- [25] Schwarz, J.: *Automobily škoda Octavia II*. První vydání, Praha, 2006. ISBN 80-247-1141-9.
- [26] Hinner, M.: *Diagnostika Automobilových řídicích jednotek*. Praha, 2006. Bakalářská práce na ČVUT v Praze. Vedoucí bakalářské práce Jiří Novák.
- [27] Kravar, Jan.: *Diagnostika elektronických řídicích jednotek*. Praha, 2006. Bakalářská práce na ČVUT v Praze. Vedoucí bakalářské práce Jiří Novák.
- [28] Datasheet SI 9243 AEY

Seznam příloh

- Příloha č. 1 Typová laboratorní úloha
- Příloha č. 2 Schéma, osazovák (TOP/BOT) a deska plošných spojů
- Příloha č. 3 Průzkum integrovaných obvodů pro realizaci CAN Bus jednotky.
- Příloha č. 4 Průzkum integrovaných obvodů pro realizaci LIN Bus jednotky.
- Příloha č. 5 Průzkum integrovaných obvodů pro realizaci FlexRay Bus jednotky
- Příloha č. 6 Vypracovaná laboratorní úloha (přiloženo v deskách DP)

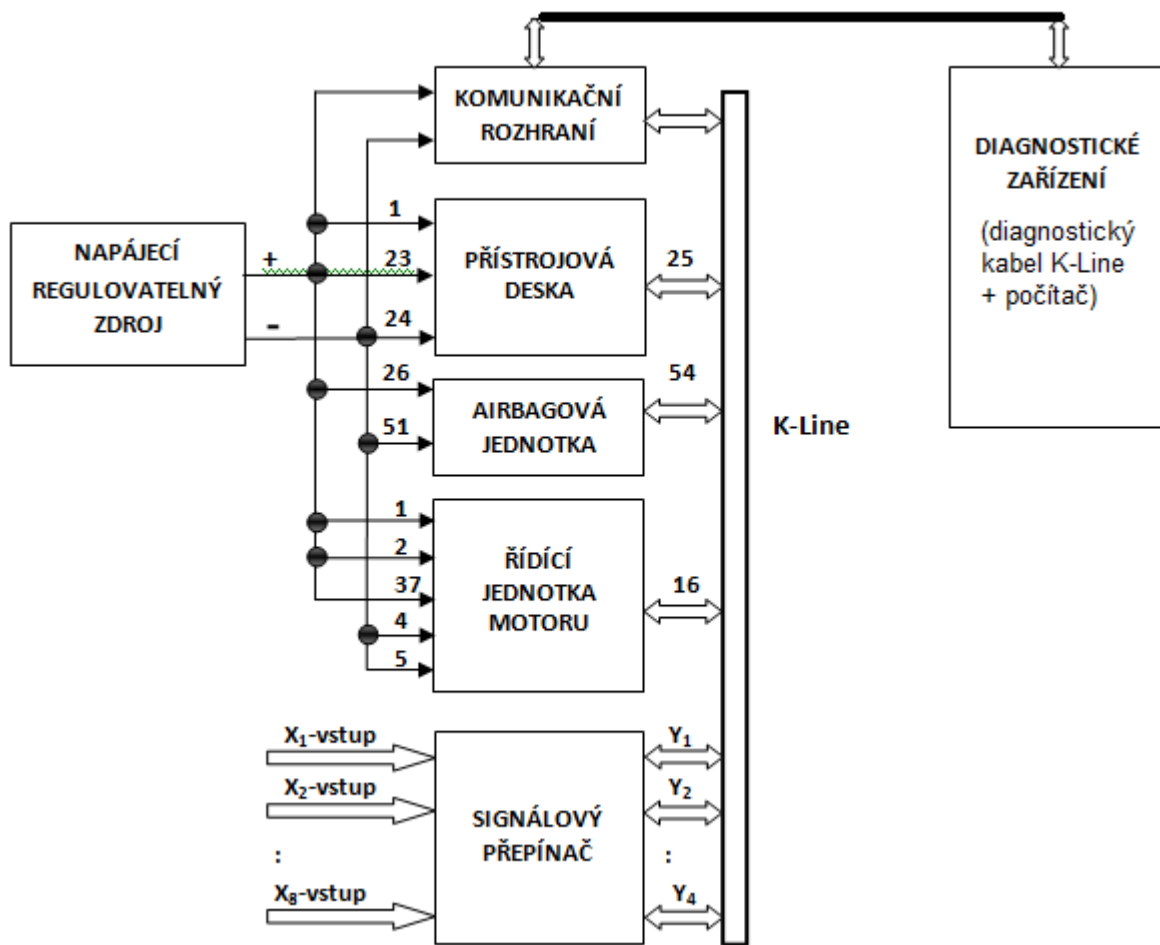
PŘÍLOHA Č. 1

Typová laboratorní úloha

VŠB-TU Ostrava	AUTOMOBILOVÁ ELEKTRONIKA		Fakulta elektrotechniky a informatiky
<i>Datum měření:</i>	1.	SÉRIOVÁ DIAGNOSTIKA, KOMUNIKACE S ŘÍDÍCÍMI JEDNOTKAMI PO VEDENÍ K- Line	<i>Jména, studijní skupiny:</i>
<i>Hodnocení:</i>			

Zadání:

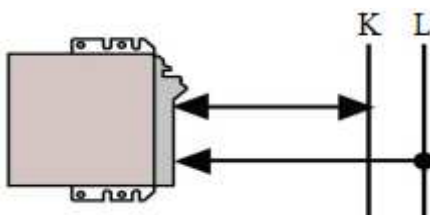
1. Zapojte diagnostický kabel K-Line do diagnostické zásuvky automobilu (EOBD), nebo proveďte zapojení řídicích jednotek na panelu dle schématu (obrázek 1) a zapojte diagnostický kabel K-Line.
2. V programu VAG – COM proveďte nastavení rozhraní a následně proveďte test komunikace, případně test LED diod. Dále otestujte kompatibilitu se systémem EOBD/OBDII.
3. Navažte komunikaci se všemi řídicími jednotkami a vyčtěte paměť závad jednotlivých systémů. Vyjádřete se k povaze chybových kódů.
4. Navažte komunikaci s jednotkou motoru a zaznamenejte pomocí funkce „Protokol LOG“ tři libovolné veličiny, které následně pomocí programu Excel zpracujte a vytvořte jeden graf, který bude obsahovat všechny zvolené veličiny. Zvolené veličiny měřte po dobu 30 sekund.
5. Zjistěte frekvenci na sběrnici K-Line a vyjádřete se k přenosové rychlosti a amplitudě komunikačního signálu.



Obrázek 1: Blokové schéma zapojení vodičové struktury K-Line

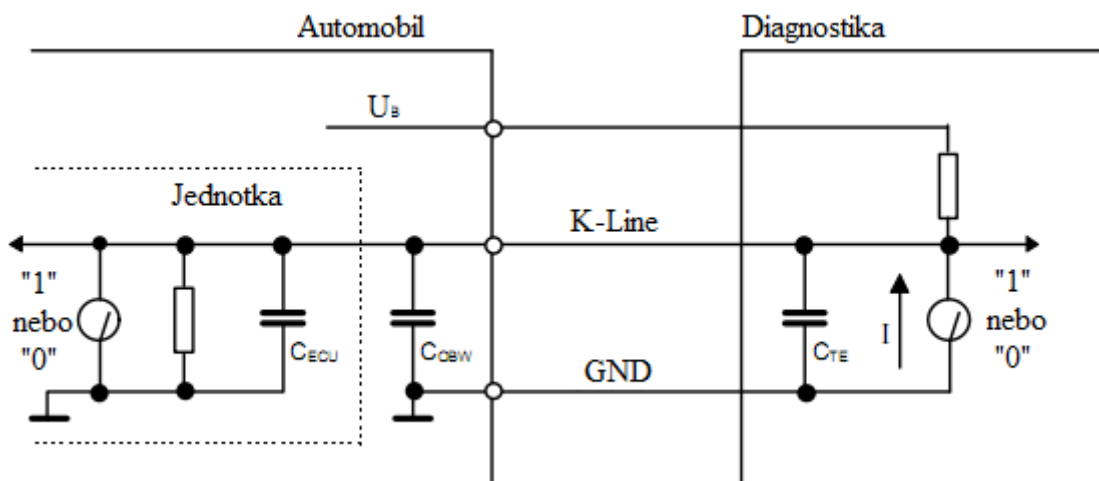
Teoretický rozbor:

Vedení K – Line a L- Line jsou diagnostické kanály, které definuje normou ISO 9141 – 2. V závislosti na roku vzniku automobilové řady se může setkat s kombinací K i L linky u starších automobilů, od roku 1996 se vyskytuje jenom K vedení a u nejnovějších modelů se lze setkat s KK diagnostickým vedením. Z obrázku 2 je patrné, že K linka komunikuje obousměrně a slouží pro posílání adres a dat. L vedení slouží pro souběžné (s K – Line) odesílání adres při inicializaci.



Obrázek 2: Směr komunikace obou linek

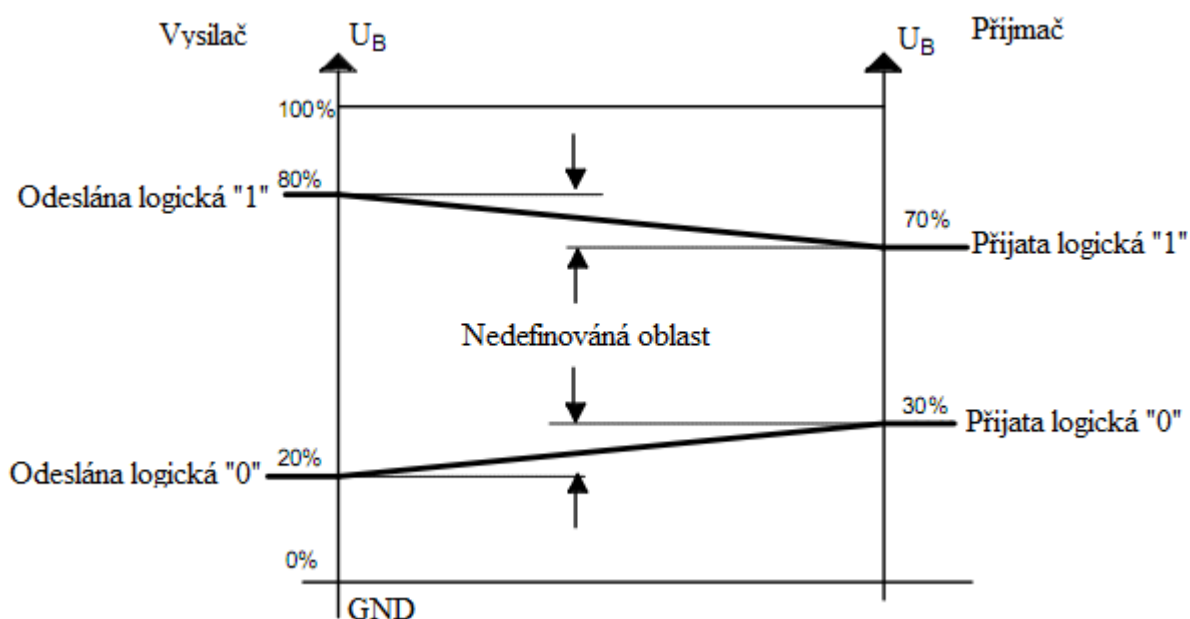
Zjednodušené zapojení jednotky a testovacího zařízení je vidět na obrázku 3.



Obrázek 3: Zjednodušené komunikační schéma podle ISO 9141

Výstupní napěťové úrovně se odvíjí od napětí na baterie U_B vůči kostře.

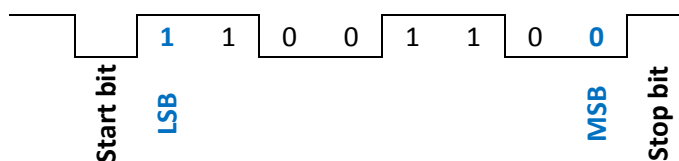
- ✓ Logické „0“ odpovídá menšímu napětí než 20% z napětí baterie pro vysílač a pro přijímač menšímu než 30% z napětí baterie.
- ✓ Logická „1“ odpovídá většímu napětí než 80% z napětí baterie pro vysílač a pro přijímač většímu než 70% z napětí baterie.



Obrázek 4: Mezní napěťové úrovně logických stavů

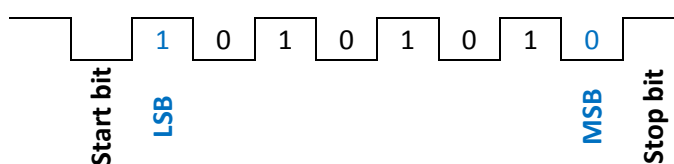
Inicializace komunikace

Podmínkou navázání spojení je udržení linky „K“ alespoň 2ms v jedničkovém stavu. Následně se vysílá osmi bitový rámec adresy s přenosovou rychlostí 5bit/s (jedná se pomalou inicializací) po „K“ a „L“ vedení. Hodnota adresového rámce je 33_H a je posílána od nejvíc významného bitu MSB po nejméně významný bit LSB. Situace je znázorněná na obrázku 5.



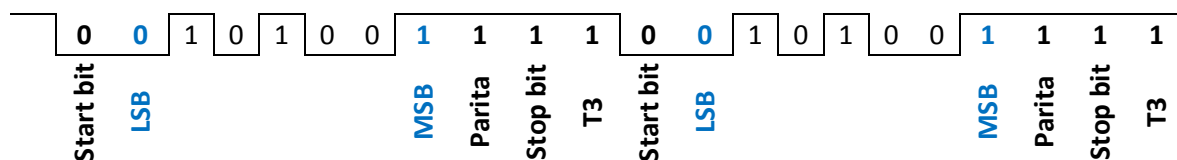
Obrázek 5: Adresový rámec

Po uplynutí času T1 (stav logická jedna), který nastane po adresovém rámci, je očekáván synchronizační bit z OBD II systému. Struktura rámce je naznačena na obrázku 6, přičemž se skládá ze střídajících se vysokých a nízkých stavů K linky při rychlosti 10,4kbit/s. Dodatečný čas na přizpůsobení rychlosti testovací jednotky je v protokolu zabezpečeno setrváním v jedničkovém logickém stavu po čas T2, za kterým OBD II systém odesílá následující bajty pro navázání spojení.



Obrázek 6: Synchronizační bajt

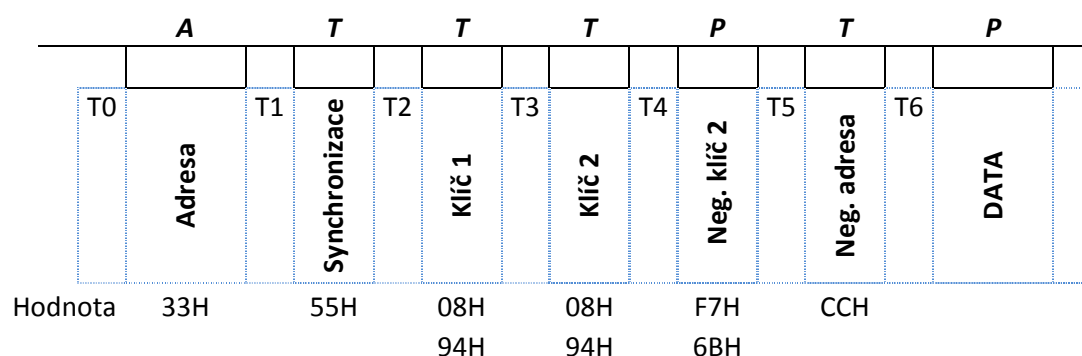
Následující bajty odesílá rovněž systém OBD II. Jedná se o dva 7 bitové rámce, označované literaturou [10], jako klíčová slova (*keyword*). Tyto klíčová slova slouží pro rozpoznání komunikačního protokolu a typu jednotky. Jsou znázorněny na obrázku 7. Ještě je třeba zmínit, že každé 7 bitové slovo je doplněno paritním bitem. Klíčová slova jsou oddělena jedničkovým bitem, který trvá čas T3.



Obrázek 7: Formát klíčových slov

Poznámka: Literatura [10] uvádí, že normalizační výbor FAKRA, sídlící v Německu, přiděluje výrobcům automobilu unikátní klíčová slova. Například čísla od 71 do 120 jsou přiděleny automobilce Opel.

Tester po přijetí druhého klíčového slova a uplynutí jedničkového stavu, který trvá čas T4, odesílá negované druhé klíčové slovo jednotce automobilu. Následně se musí udržet jedničkový stav po dobu T5. A pak je odeslána systémem OBD II negovaná hodnota adresy (CC_H). Struktura celé sekvence je znázorněna na obrázku 8.



*Směr komunikace: A – automobil, T – Tester

Obrázek 8: Inicializace přenosu v systému ISO 9141 – 2

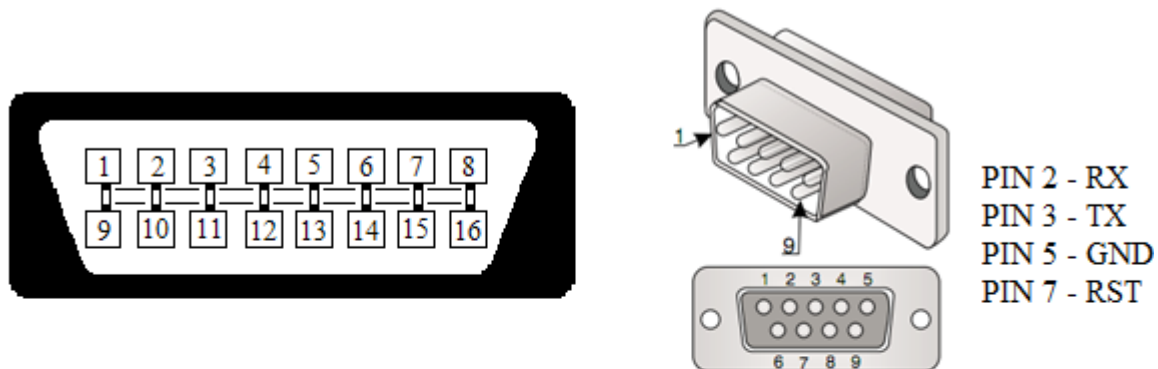
Datový rámeček

Komunikace s ohledem na přenos diagnostických dat je spojena v pakety, které jsou nazývány informacemi. Skládají se z 5 až 11 bajtů, přehled je znázorněn na obrázku 9. K obrázku je třeba doplnit, že jednotlivé datové informace jsou oddělené časem P2 a P3 v závislosti na směru přenosu dat (do automobilu / testeru). A jednotlivé bajty v datové informaci, jsou odděleny dobou P1 (bajty z automobilu) a P4 (bajty z testeru).

Hlavička			Datové pole							Kontrolní bajty
Priorita	Cílová adresa	Zdrojová adresa	1	2	3	4	5	6	7	ERR
<i>Diagnostický požadavek při přenosové rychlosti 10,4kbit/s podle ISO 9141 - 2, SAE J1850</i>										
68h	6Ah	FXh	Datové pole (maximálně 7 bajtů)							ERR
<i>Diagnostická odpověď při přenosové rychlosti 10,4kbit/s podle ISO 9141 - 2, SAE J1850</i>										
48h	6Bh	drojová adres	Datové pole (maximálně 7 bajtů)							ERR
<i>Diagnostický požadavek při přenosové rychlosti 10,4kbit/s podle ISO 14230 - 4</i>										
11DDDDDDb	6Ah	FXh	Datové pole (maximálně 7 bajtů)							ERR
<i>Diagnostická odpověď při přenosové rychlosti 10,4kbit/s podle ISO 14230 - 4</i>										
10DDDDDDb	FXh	drojová adres	Datové pole (maximálně 7 bajtů)							ERR
ERP - bajt kontrolní sumy CRC										
DDDDDD - délka dat (jen ISO 14230 - 4)										

Obrázek 9: Datové diagnostické formáty v systému OBD II / EOBD

Zapojení EOBD a DB9 konektoru



Obrázek 10: Konektor EOBD a RS232 (DB9)

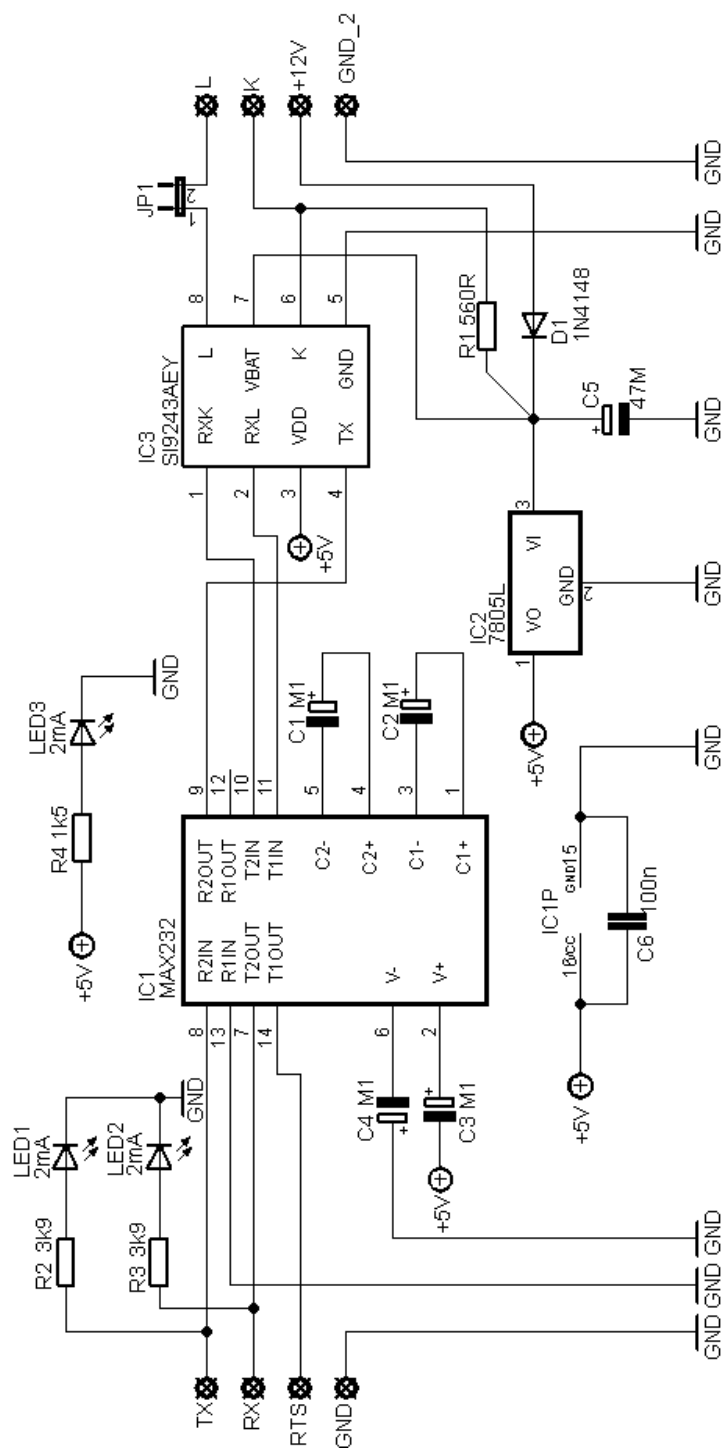
Obsazení pólů diagnostického konektoru - zásuvka EOBD

1, 3, 8, 9, 11, 12, 13 – neobsazeno	7 – K-Line
2 – komunikační protokol SCP +	10 – komunikační protokol SCP -
4 – kostra vozidla	14 – CAN <u>Low</u>
5 – kostra signálu	15 – L-Line
6 – CAN <u>High</u>	16 – +12V

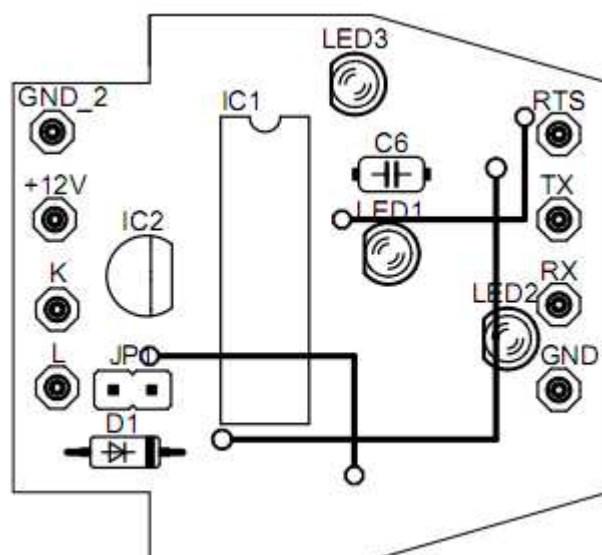
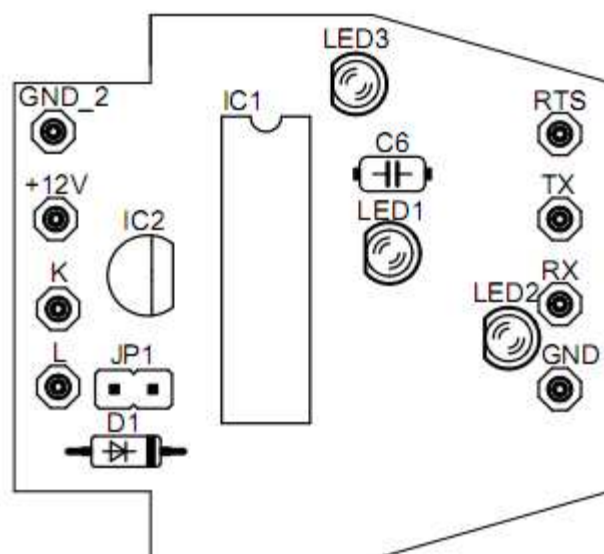
Poznámka: Protokol je vypracován a založen v deskách DP

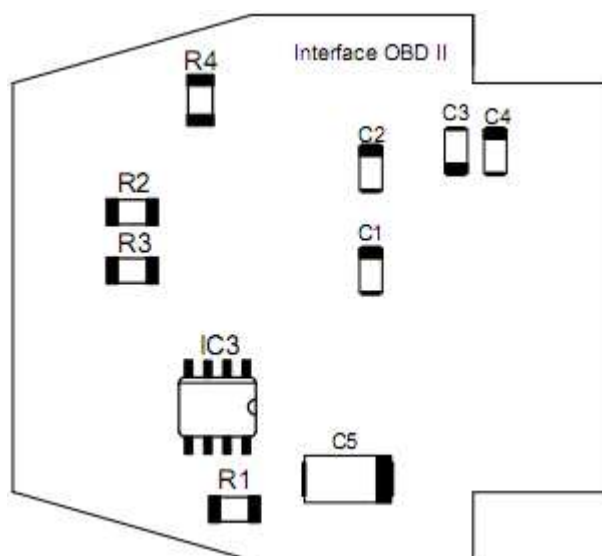
PŘÍLOHA Č. 2

Schéma, osazovák a návrh DPS

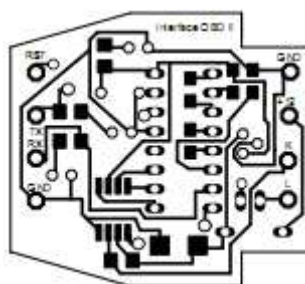


Obrázek 1: Schéma K – Line interface





Obrázek 4: Osazovák BOT strany



Obrázek 5: Deska plošných spojů

Soupiska potřebných součástek

Označení	Hodnota	Pouzdro	Poznámka
C1	M1	SMC_A	tantalový/10V
C2	M1	SMC_A	tantalový/10V
C3	M1	SMC_A	tantalový/10V
C4	M1	SMC_A	tantalový/10V
C5	47M	SMC_C	tantalový/16V
C6	100n	C050	slitkový
D1	1N4148	DO35-7	
IC1	MAX232	DIL16	
IC2	7805L	TO92	
IC3	SI9243AEY	SO08	
JP1		JP1	

LED1	2mA	LED3MM
LED2	2mA	LED3MM
LED3	2mA	LED3MM
R1	560R	R1206
R2	3k9	R1206
R3	3k9	R1206
R4	1k5	R1206

RX	TX	ZAP/VYP
TOK DAT		STAV

Obrázek 6: Popisek Led diod interface



Obrázek 7: Fotografie vyrobeného Hardware

PŘÍLOHA Č. 3

PRŮZKUM INTEGROVANÝCH OBVODŮ PRO REALIZOVÁNÍ CAN BUS

Výrobce IO	Typ IO	Popis IO
Philips	AN97076	CAN řadič SJA1000 (CAN 2.0A , CAN 2.0B)
	PCA82C250	Transceiver High-Speed CAN
	PCA82C252	Transceiver Low-Speed CAN
Freescale	MC33388	Transceiver Low-Speed CAN
	MC33989	Transceiver High-Speed CAN
	MC33742	Transceiver High-Speed CAN/Diagnostics
Atmel	AT90CAN128	8-bitový mikrokontrolér architektury AVR, RISC
	AT89V51CC03	mikrokontrolér architektury 8051
	ATmega16M1	16-bitový mikrokontrolér
	ATmega32M1	32-bitový mikrokontrolér
	ATmega64M1	64-bitový mikrokontrolér
Intel	82527	16-bitový mikrokontrolér
NXP Semiconductor	TJA1040	Transceiver High-Speed CAN
Microchip	MCP2510	CAN řadič (CAN 2.0A , CAN 2.0B)
STMicroelectronics	L9669	Transceiver Fault-tolerant CAN

PŘÍLOHA Č. 4

PRŮZKUM INTEGROVANÝCH OBVODŮ PRO REALIZOVÁNÍ LIN BUS

Výrobce IO	Typ IO	Popis IO
Fujitsu	MB89210 série	8-bitový mikrokontrolér s podporou LIN
	MB90390 série	16-bitový mikrokontrolér s podporou LIN
	MB91F364G	32-bitový mikrokontrolér s podporou LIN
Microchip	MCP201	LIN transceiver se stabilizátorem napětí
	PIC16C432	8-bitový mikrokontrolér s LIN transceiverem
	PIC16C433	8-bitový mikrokontrolér s LIN transceiverem
Motorola	MC33399	realizace fyzické vrstvy LIN
	MC33895	čtyřnásobný poloviční můstek
	MC68HC908	8-bitový mikrokontrolér
	MC9S12C32	16-bitový mikrokontrolér
	MC9S12D64	16-bitový mikrokontrolér
Melexis	TH8060	LIN transceiver se stabilizátorem napětí
	TH8080	LIN transceiver
	TH8100	LIN slave kontrolér pro spínače
Philips	TJA1020	LIN transceiver
Infineon	TLE 6258G	LIN transceiver
Amis	AMIS-30600	LIN transceiver

PŘÍLOHA Č. 5

PRŮZKUM INTEGROVANÝCH OBVODŮ PRO REALIZOVÁNÍ FLEXRAY BUS

Výrobce IO	Typ IO	Popis IO
Philips	TJA 1080	Budič FlexRay, bez hlídacího členu
	ARM9 SJA2510	32 - bitový procesor s řadičem FlexRay 2.1
Austriamicrosystems	AS82xx	Budič FlexRay, bez hlídacího členu
Freescale	MFR4310	Řadič FlexRay, verze 2.1
	MC9S12XF	Procesor s integrovaným FlexRay řadičem 2.1
	MFR4200	Řadič FlexRay, verze 1.1
Fujitsu	MB88121	Budič FlexRay, bez hlídacího členu
	MB91F465x	Procesor s integrovaným FlexRay řadičem 2.1
Fujitsu	MB88121	Budič FlexRay, bez hlídacího členu